DATA ACQUISITION BASED ON COMMODORE 64

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

by

Kamyab Aghai-Tabriz

In Partial Fulfillment of the Requirements
for the Degree of
Master of Science

Major Department: Mechanical Engineering and Applied Mechanics

May, 1985
Fargo, North Dakota

This thesis submitted by Kamyab Aghai-Tabriz in partial fulfillment of the requirements for the Degree of Master of Science from North Dakota State University is hereby approved by the Faculty Advisory Committee under whom the work has been done.

K. G. Maurer, Mechanical Engineering and Applied Mechanics
Department Chairman

Kyriakos Okamura Major Advisor

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Dr. Kiyohisa Okamura for his his guidence and encouragement throughout the preparation of this thesis. Special thanks to professor Sakshaug for helping and his constructive criticism. In particular, the author would like to express his appreciation to all the committee members.

Thanks are also due to all the Mechanical Engineering faculty and staff.

TABLE OF CONTENT

ACKNOWLEDGEMENT iii

Chapter page

I. INTRODUCTION 1

II. OVERVIEW OF TOTAL SYSTEM 5

III. HARDWARE 7

Transducers and Sensors 7
Amplifiers 7
Multiplexer (MUX) 11
Analog to Digital Conversion (ADC) 12

IV. SOFTWARE 17

Machine language subroutines 20
Data collection 20
Plotting data on high-resolution screen 26
Time mode 27
X-Y mode 27
High-Resolution screen to printer 32
BASIC subroutines 35
Storing and Recalling data 35
Transmission 36

V. THE EXPERIMENTS 37

Cam analysis experiment 38
calibration of velocity transducer 39
method of calibration 42
Procedure and results of cam experiment 45
Air compressor experiment 52
Calibration of pressure transducers 54
Calibration of the amplifiers 55
Volume calculation 55
procedure and results 61
Industrial application: Consulting project 69

VI. DATA TRANSMISSION 76

Transmission to desktop computer 76
Voltage matching between RS-232 s 76
Software for transmission 79

Procedure for transmission format 79
Transmission to main frame computer 83

VII. EVALUATION OF THE SYSTEM AND CONCLUSION 84

Limitation of the system 86
Applications 87
Outlook for the DAS 88

Appendix page

A. CALIBRATION METHOD FOR VELOCITY TRANSDUCER . . . 89

B. SOFTWARE FOR DAS (BASIC) 94

C. ADC ML PROGRAM 111

D. HIGH-RESOLUTION PLOTTING ROUTINE 115

Clear High-Resolution memory 127
Color assignment 127
Find channel numbers 127
High-Resolution: One channel 128
High-Resolution: Two channels 128
High-resolution: Three channels 129
Plot the adjusted values 130

E. HIGH-RESOLUTION SCREEN TO PRINTER ROUTINE 133

F. AMPLIFIERS DATA 137

G. PRESSURE TRANSDUCER DATA 144

H. USER MANUAL FOR COMPRESSOR 148

I. USER MANUAL FOR GENERAL DATA ACQUISITION 162

Take data in 165
Plot on screen 167
Graph on printer 168
Transmit data 169
Store data 170
Recall data 171

J. USER MANUAL FOR TRANSMISSION 173

BIBLIOGRAPHY 177

LIST OF FIGURES

Figure		page
2.1.	Data acquisition system	6
3.1.	Block diagram For Data Acquisistion System (DAS)	8
3.2.	Amplifiers AD522 circuit diagram	10
3.3.	Multiplexer 4051 circuit diagram	14
3.4.	Analog To Digital Converter ADC0804 circuit diagram	15
3.5.	Complete circuit diagram for DAS	16
4.1.	Memory map of Data Acquisition System	18
4.2.	General flowchart for Data Acquisition System	19
4.3.	Data collection routine flowchart	22
4.4.	Timing diagram for Analog to Digital Converter chip	23
4.5.	Test signal collected by DAS before interrupt disabled	25
4.6.	Test signal collected by DAS after interrupt disabled	25
4.7.	Horizontal and vertical pixels in C-64 Hi-Res	26
4.8.	Bytes and bits arrangement of C-64's High-Resolution screen	28
4.9.	(A) Flowchart for the plotting routine (time mode)	29
4.10.	(B) Flowchart for the plotting routine(time mode)	30
4.11.	Flowchart for the plotting routine (X-Y mode)	31
4.12.	Flowchart of the printer routine	34
4.13.	Format of data storage on the disk	36
5.1.	Connection of DAS for velocity transducer calibration	40
5.2.	Plot of velocity data (MSP-801 printer)	41

5.3. Velocity transducer equivalent circuit 42
5.4. Manufacturer calibration curve for velocity transducer . . . 43
5.5. Velocity output for calibration 44
5.6. Block diagram for cam experiment 47
5.7. Displacement of the cam 48
5.8. Velocity of the cam 49
5.9. Acceleration of the cam 50
5.10. The cam profile 51
5.11. Engine indicator used to plot CPI 53
5.12. Photo 1 low and high pressure data and time marker 56
5.13. (A) Place and Position of photo transistor (B) Circuit diagram for photo transistor 57
5.14. The flywheel and the position of photo transistor 58
5.15. The configuration used for volume calculation 60
5.16. PT diagram for low pressure stage 62
5.17. PT diagram for high pressure stage 63
5.18. CPI diagram for low pressure stage 64
5.19. CPI diagram for high pressure stage 65
5.20. CPI diagram for the compressor (both stage) 66
5.21. PT and PV diagram for a compressor (Adopted from Mechanical Engineering Magazine, DEC 1984 ,Page 67.) 68
5.24. The coffee pot and the position of thermocouples 70
5.25. thermocouple 1 output against time (wet test) 72
5.26. Thermocouple 2 output against time (wet test) 73
5.27. Thermocouple 3 output against time (wet test) 74
5.28. Thermocouple 1 and 2 outputs against time (dry test) . . . 75
6.1. The line driver MC1488 implementation 78

A.1. Experimental setup 89

A.2. Calibration curve 92

B.1. First screen display 99

B.2. Display of main menu 100

B.3. Main menu for the compressor data acquisition 107

B.4. A sample of pressure transducer and photo transistor output 108

B.5. Sample plot of the PV diagram on high resolution screen 110

F.1. Calibration curve for amplifier No. 1 (gain=101.67) 138

F.2. Calibration curve for amplifier No. 1 (gain=501.49) 139

F.3. Calibration curve for amplifier No. 1 (gain=1000.65) . . . 140

F.4. Calibration curve for amplifier No. 2 (gain=102.09) 141

F.5. Calibration curve for amplifier No. 2 (gain=504.34) 142

F.6. Calibration curve for amplifier No. 2 (gain= 1015.25) . . . 143

G.1. Calibration curve for pressure transducer No. 1 145

G.2. Calibration curve for pressure transducer No. 2 146

H.1. Commodore 64 and peripheral 157

H.2. Connection for Commodore 64 and DACQ1 158

H.3. Pressure Transducer- DACQ1 connection (1) 159

H.4. Pressure Transducer- DACQ1 connection (2) 160

H.5. Signal flow block diagram for DACQ1 and Commodore 64 . 161

I.1. Main menu of DAS software 166

I.2. Sub-menu for channel selection 168

I.3. Sub-menu for sampling rate selection 169

I.4. Sub-menu for data transmission 170

I.5. Sub-menu for data transmission 170

I.6. Sub-menu for data storage 171

LIST OF TABLES

Table		page
3.1.	Truth table of MUX 4051	11
5.1.	Result of cross calibration	41
5.2.	Cross calibration using calibrated factor	45
5.3.	Result of RPM calibration	59
F.1.	Amplifiers gain	137

Chapter I
INTRODUCTION

A special news release by the Accreditation Board for Engineering and Technology (ABET) in August,1983,stated that the

> "...The laboratory condition in the U.S. institution of higher learing is deteriorating and near-term disaster would result from continuing present trends...."

The Applied High-Tech Laboratory of the MEAM Department has been responsible for the general instrumentation for the department laboratories and must conform to the ABET requirements.

According to Dr. K. Okamura, Associate Professor of Mechanical Engineering Department, the installation of integrated data acquisition systems in the MEAM laboratories is among the top priorities in conjuction with the ABET guidelines. A few specialized data acquisition systems have been developed and are in use in the Applied High-Tech Laboratories. However, it is desirable to install the departmental interlaboratory network of data acquisition, transmission and processing systems for general use. Toward this goal the coordinated effort started under Dr. Okamura and the author.

The objective of this thesis was to design and construct a low cost Data Acquisition System (DAS) and apply the system to the existing laboratory experimental apparatus. The requirements for this thesis project were:

1. With a limited equipment budget, the system must be inexpensive enough for the Mechanical Engineering Department to purchase and install a setup for each experimental station.
2. The system must be flexible enough that both hardware and software can be adapted for each experimental apparatus. The range of flexibility should include: capability of storing the information, immediate display of results, some numerical analysis and numerical processes, and transmission to a microcomputer or the main frame computer.

A commercially available multi-channel data acquisition system would cost more than ten thousand dollars. Even an adapter unit for some of the well known personal computers costs a few hundred dollars in addition to the cost of the computer itself.

The author's pilot study proved Commodore 64 home computer is adequate and can meet the above requirements. Some of the major advantages of this computer are:

1. Because of the high volume production (the most popular microcomputer in the U.S.A.), the unit price is among the lowest;
2. Well developed peripherials are available at low cost;
3. Proper I/O (Input/Output) ports are available;
4. High resolution display is provided at a reasonable cost;
5. Various software in ROM (Read Only Memory) or disk are available which can be used for data acquisition, display and transmission.

The main purpose was to design and construct a flexible and adaptable prototype data acquisition system. The data acquisition system

based on C-64 computer with the above specification was designed, built and successfully tested. The system was designed to gather information from transducers installed on each apparatus. This was done by amplifying and conditioning the electrical signals from transducers located at each site. The electrical signals were then converted to digital values and stored in the memory of the computer.

The 8K[1] RAM (Random Access Memory) location allocated for data storage was adequate for all the experiments. After each experimental session, the data stored on the disk was transmitted either (1) from the experimental site directly through co-axial cable to a larger desk top computer at a 2400 baud rate (TRS80 Model II, manufactured by Radio Shack, a division of Tandy Corporation) or (2) by use of modem and telephone line directly to NDSU main frame computer at a 300 baud rate (two IBM 4341's running in parallel with OS/MVS2/SP operating system (IBM 370)) where data analysis and plots were done. The maximum sampling rate of the system is 4360 data per second which is adequate for most mechanical analysis.

The system developed has many unique features not available in commercial data acquisition systems. The summary of the design of the system was published in the Feburary issue of BYTE [1].[2]

This thesis is written as follows: first, the objective and procedure of the research experiments have been given in this chapter. The main body of the report starts with an overview of the total system in chapter 2 and ends with comparison of results with known values and analy-

---

[1] 1 K memory = 2^{10} , or 1024 bytes.

[2] Numbers in brackets designate Reference at the end of the thesis.

sis in chapter 7.

The remainder of the main text separates the system and experiments and discusses each in detail. Chapter 5 provides information on cam analysis and the two-stage air compressor experiments. Chapter 6 describes the various methods of transmission of collected data from the Commodore 64 to a larger computer.

Chapter II

OVERVIEW OF TOTAL SYSTEM

Figure 2.1 is a drawing illustrating the Data Acquisition System (DAS) developed as this thesis project. A transducer (1) senses the condition of the mechanical or process system and produces corresponding proportional electrical signals. These signals are applied through a signal conditioner to a multiplexer (3). The multiplexer (MUX) makes it possible to sample the signals of many transducers. Rotary switches are used as MUX in older systems. Present-day systems are using solid-state electronic MUX to accomodate high speed switching.

The signal conditioner between transducers and MUX is simply an instrumentation amplifier (2), since, in many cases the output of transducers are of millivolts order and must be amplified to a level of volts.

An analog-to-digital converter (4) is used to convert the conditioned signals out of MUX to an eight-bit digital value. This is because the computer can only respond or understand the digital pulses. At this point the Commodore 64 (5) can read these values through software and store them in the memory.

The data collected can be displayed immediately on the CRT (6) or can be stored semi-permanently on a floppy diskette for later retrieval. Also the data can be plotted or printed on a dot matrix printer (7), or transmitted through either a co-axial cable to desktop computer or through a modem to NDSU main frame for processing and analysis.

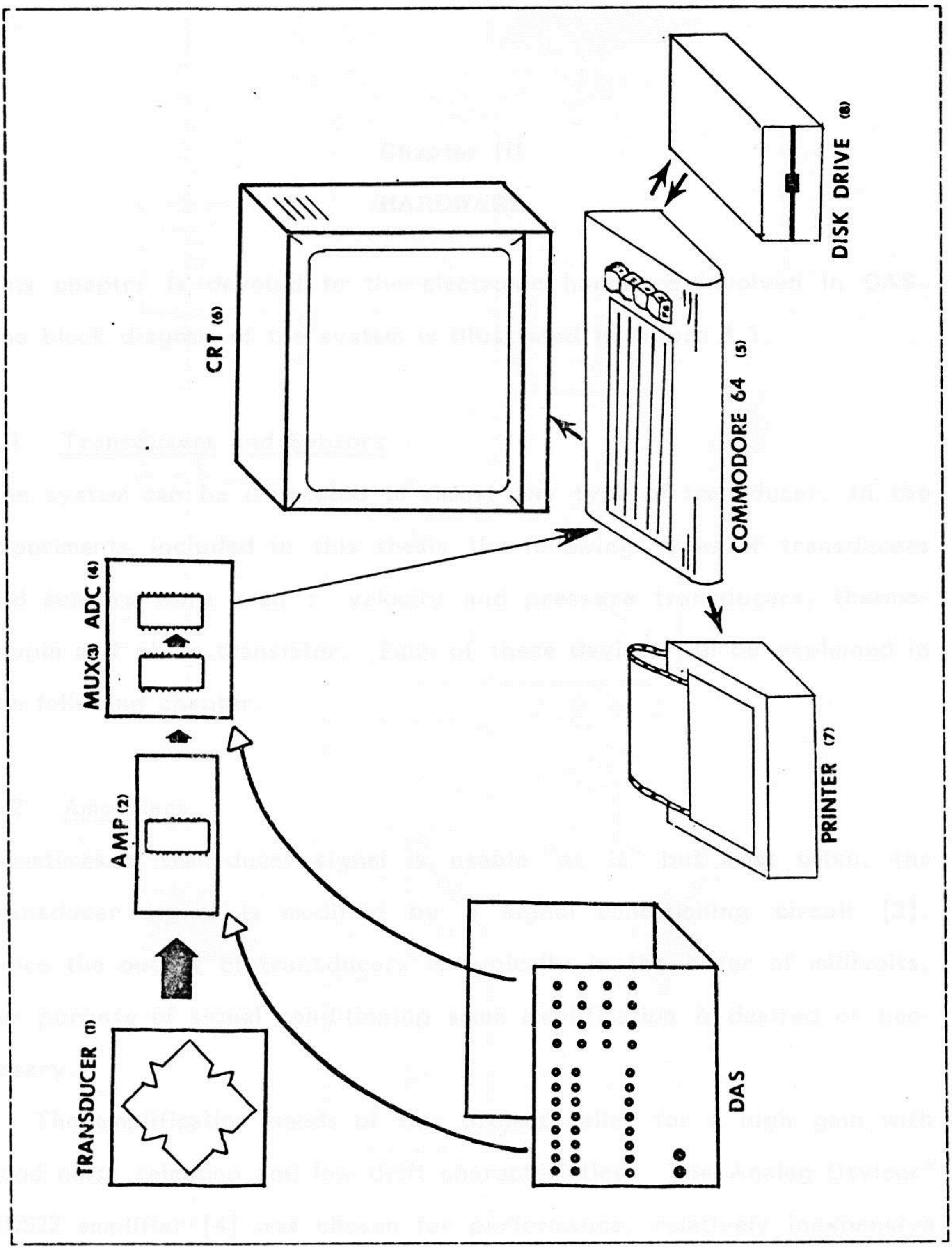


Figure 2.1: Data acquisition system

Chapter III

HARDWARE

This chapter is devoted to the electronic hardware involved in DAS. The block diagram of the system is illustrated in figure 3.1.

3.1 Transducers and Sensors

The system can be connected to almost any type of transducer. In the experiments included in this thesis the following types of transducers and sensors were used : velocity and pressure transducers, thermocouple and photo transistor. Each of these devices will be explained in the following chapter.

3.2 Amplifiers

Sometimes a transducer signal is usable "as is" but most often, the transducer signal is modified by a signal conditioning circuit [3]. Since the output of transducers is typically in the order of millivolts, for purpose of signal conditioning some amplification is desired or necessary.

The amplification needs of this project called for a high gain with good noise rejection and low drift characteristics. The Analog Devices[3] AD522 amplifier [4] was chosen for performance, relatively inexpensive cost and ease of operation. In order to reduce noise, filter capacitors

[3] Analog Devices, Inc, Norwood, MA.

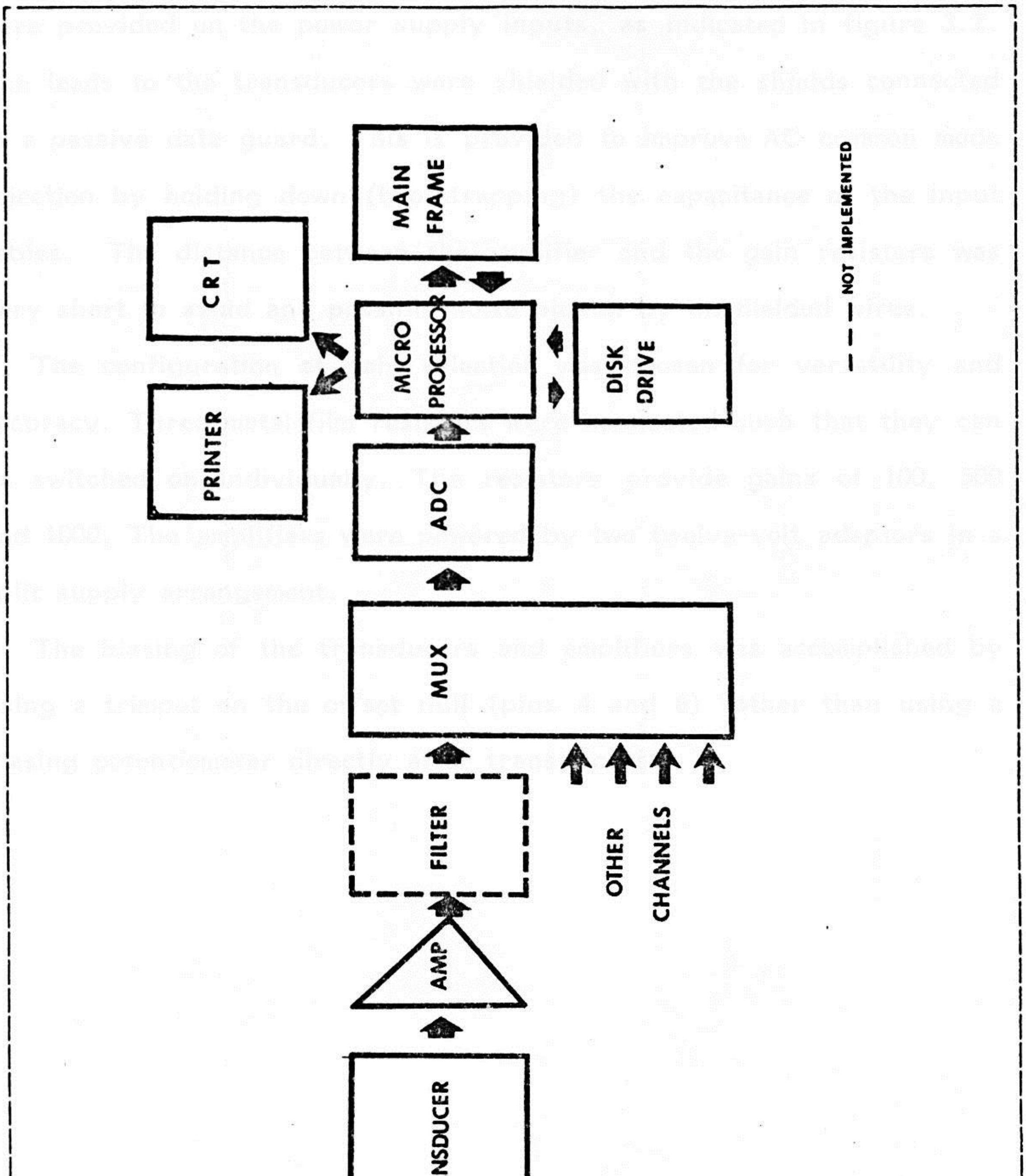


Figure 3.1: Block diagram For Data Acquisition System (DAS)

were provided on the power supply inputs, as indicated in figure 3.2. The leads to the transducers were shielded with the shields connected to a passive data guard. This is provided to improve AC common mode rejection by holding down (bootstrapping) the capacitance of the input cables. The distance between the amplifier and the gain resistors was very short to avoid any possible noise pickup by unshielded wires.

The configuration of gain selection was chosen for versatility and accuracy. Three metal film resistors were connected such that they can be switched on individually. The resistors provide gains of 100, 500 and 1000. The amplifiers were powered by two twelve-volt adaptors in a split supply arrangement.

The biasing of the transducers and amplifiers was accomplished by using a trimpot on the offset null (pins 4 and 6) rather than using a biasing potentiometer directly after transducers.

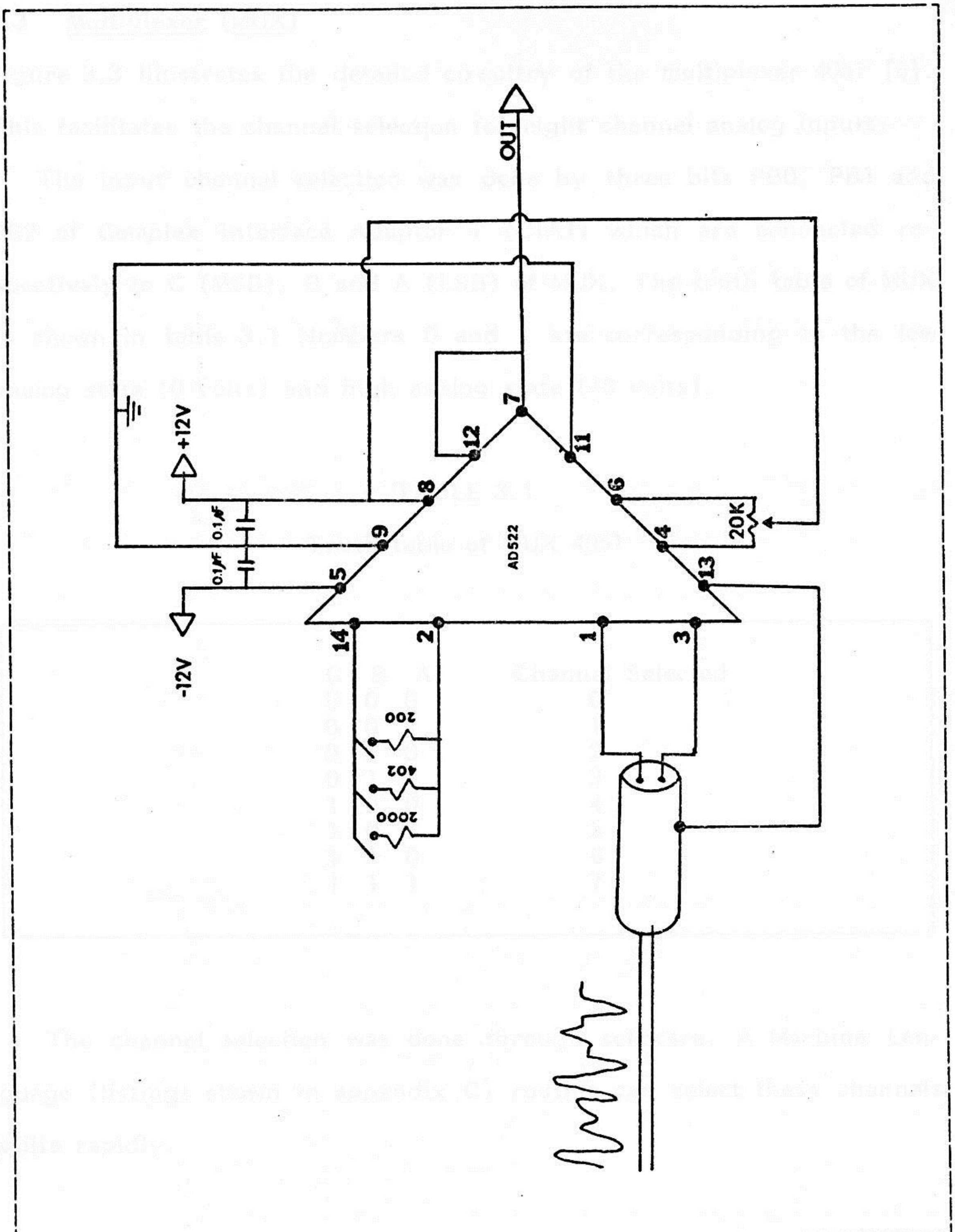


Figure 3.2: Amplifiers AD522 circuit diagram

3.3 Multiplexer (MUX)

Figure 3.3 illustrates the detailed circuitry of the multiplexer 4051 [6]. This facilitates the channel selection for eight channel analog inputs.

The input channel selection was done by three bits PB0, PB1 and PB2 of Complex Interface Adaptor 1 (CIA1) which are connected respectively to C (MSB), B and A (LSB) of MUX. The truth table of MUX is shown in table 3.1 Numbers 0 and 1 are corresponding to the low analog state (0 volts) and high analog state (+5 volts).

TABLE 3.1

Truth table of MUX 4051

C	B	A	Channel Selected
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The channel selection was done through software. A Machine Language (listings shown in appendix C) routine can select these channels quite rapidly.

3.4 Analog to Digital Conversion (ADC)

Figure 3.4 is the circuit diagram for the interfacing of ADC0804 [5] (manufactured by National Semiconductors) with Commodore 64 computer. This integrated circuit chip is capable of converting an analog input voltage to an 8-bit digital value. The analog signal for input should be in range of 0V to +5V, the 0V corresponding to 00000000 and +5V to 11111111. The Decimal equivalent of 00000000 and 11111111 is 0 and 255, respectively. Any value between these two bounds is proportionally converted to a digital value.

This chip works with successive approximation logic. The most significant bit is tested first and after 8 comparisons (64 clock cycles) a digital 8-bit binary code is transferred to the output latch [5]. The system clock used to drive the ADC is created with an external RC network. The output lines are connected to data bus PB0-PB7 of the Complex Interface Adapter 2 (CIA2) through C-64's USER PORT CN2.

The ADC0804 has four control lines for handshake with a micro-processor:

1. - ($\overline{CS}$) the chip selector is an input line which activates the chip when the line is low. Since only one chip is used and the system is in continuous conversion mode this line was connected to the system ground(low).
2. - ($\overline{RD}$) the read line is an input line that enables the output latches and allows the output to be sent to micro-processor. This is also connected to system ground. This means that as soon as the ADC is done with conversion the converted data is transferred to output lines.

3. - ($\overline{\mathrm{INTR}}$) The interrupt is an output line indicating when the conversion process is complete. This was not implemented on this system since the ML language takes more time than ADC. Hence, every time the processor looks for new data, the previous data is replaced by the current data.

4. - ($\overline{\mathrm{WR}}$) The write is an input line to signal the ADC to start conversion process. This line is connected to $\overline{\mathrm{PC2}}$ of C-64's userport CN2 [7]. $\overline{\mathrm{PC2}}$ is normally at a high state until the data bus register is read. Consequently, the processor sends a low pulse after one clock cycle.

As noted above, only one control line is implemented for data acquisition, thus making the circuit diagram very simple. The accuracy of the ADC directly depends upon accuracy and stability of the voltage supplied to REF/2 (pin9). Figure 3.5 illustrates the interconnection of all the hardware described in this chapter.

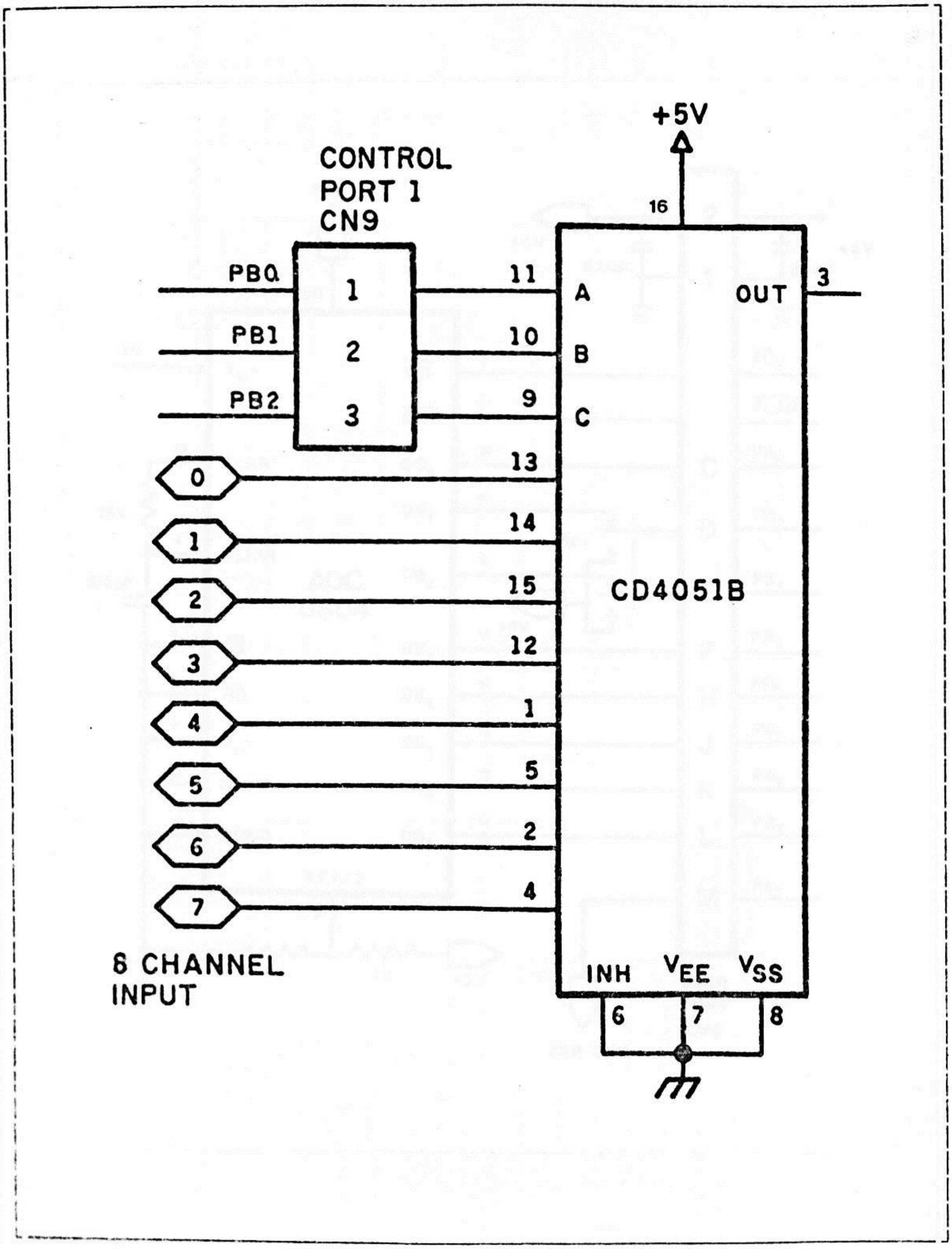


Figure 3.3: Multiplexer 4051 circuit diagram

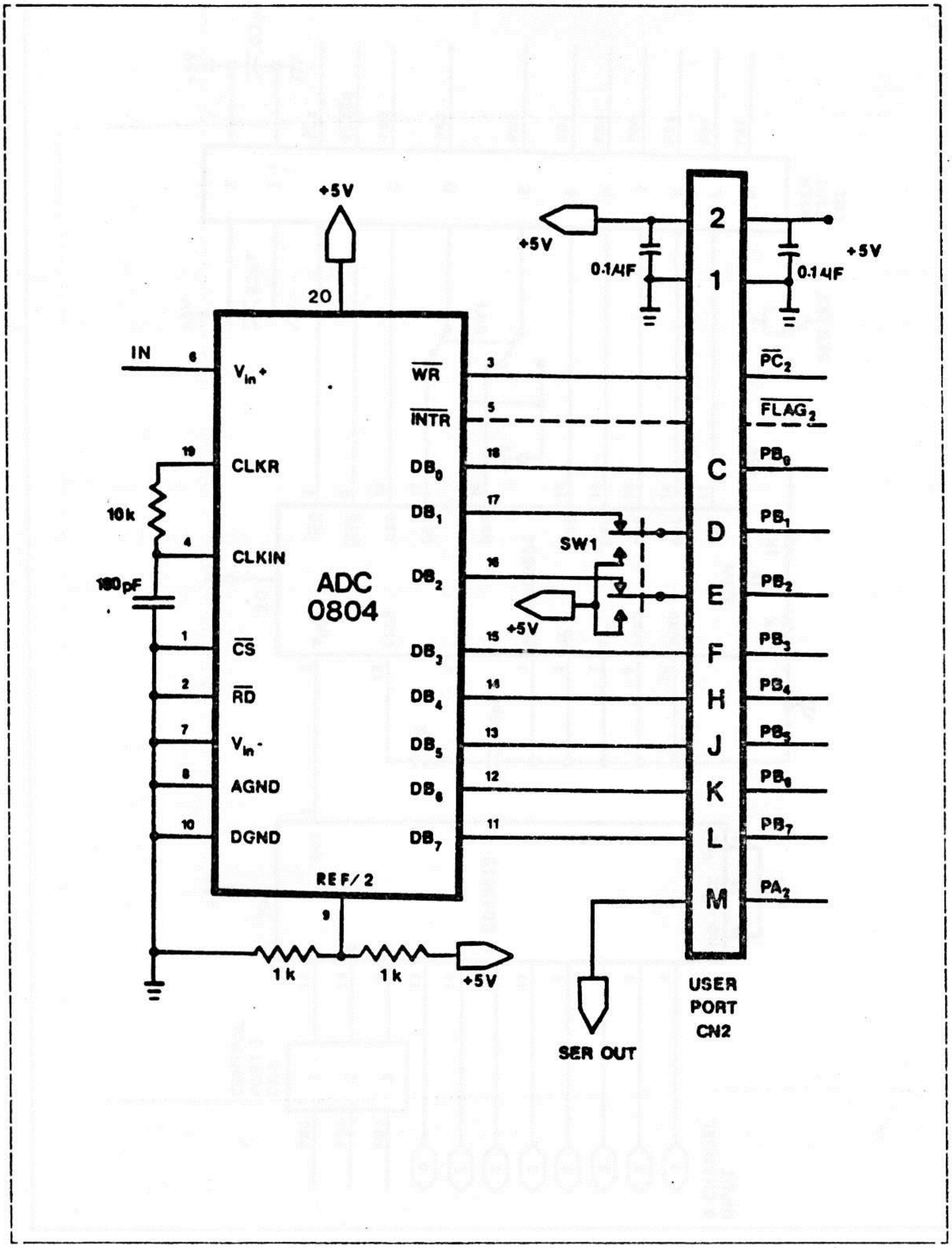


Figure 3.4: Analog To Digital Converter ADC0804 circuit diagram

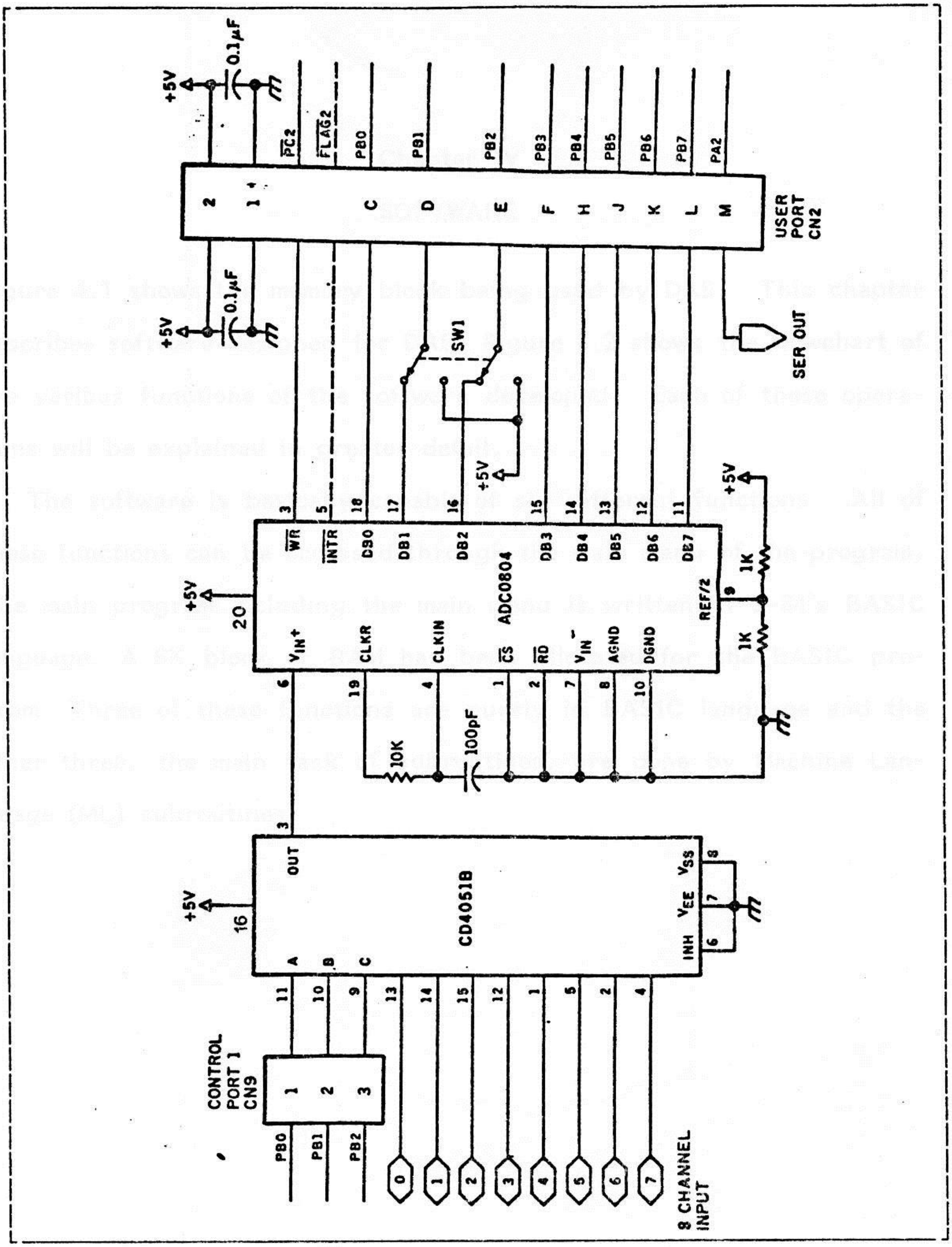


Figure 3.5: Complete circuit diagram for DAS

Chapter IV

SOFTWARE

Figure 4.1 shows the memory block being used by DAS. This chapter describes software designed for DAS. Figure 4.2 shows the flowchart of the various functions of the software developed. Each of these operations will be explained in greater detail.

The software is basically capable of six different functions. All of these functions can be accessed through the main menu of the program. The main program including the main menu is written in C-64's BASIC language. A 6K block of RAM has been allocated for the BASIC program. Three of these functions are purely in BASIC language and the other three, the main task of subroutines, are done by Machine Language (ML) subroutines.

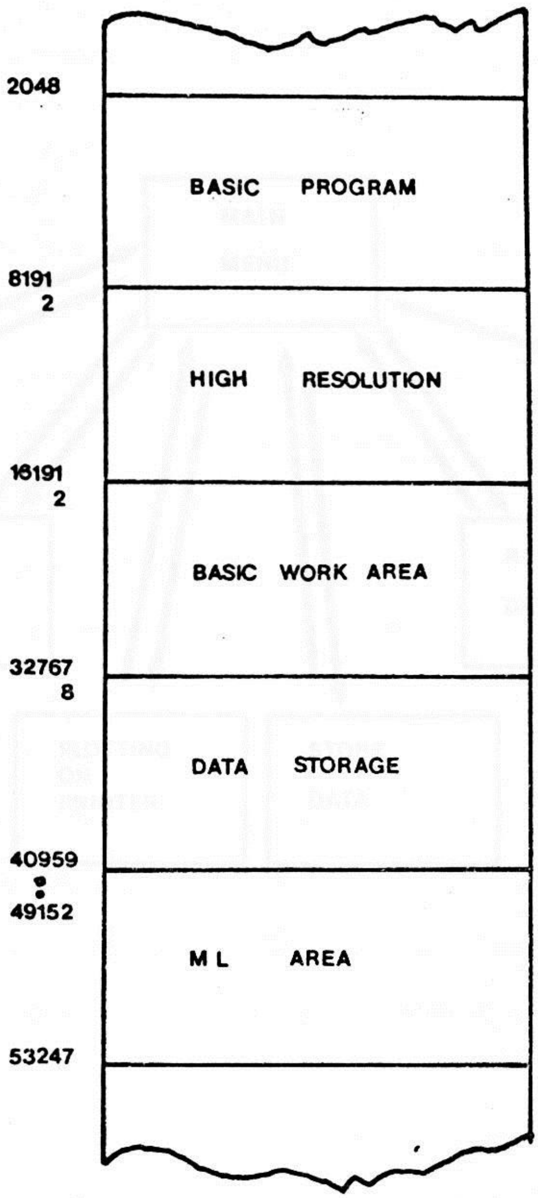


Figure 4.1: Memory map of Data Acquisition System

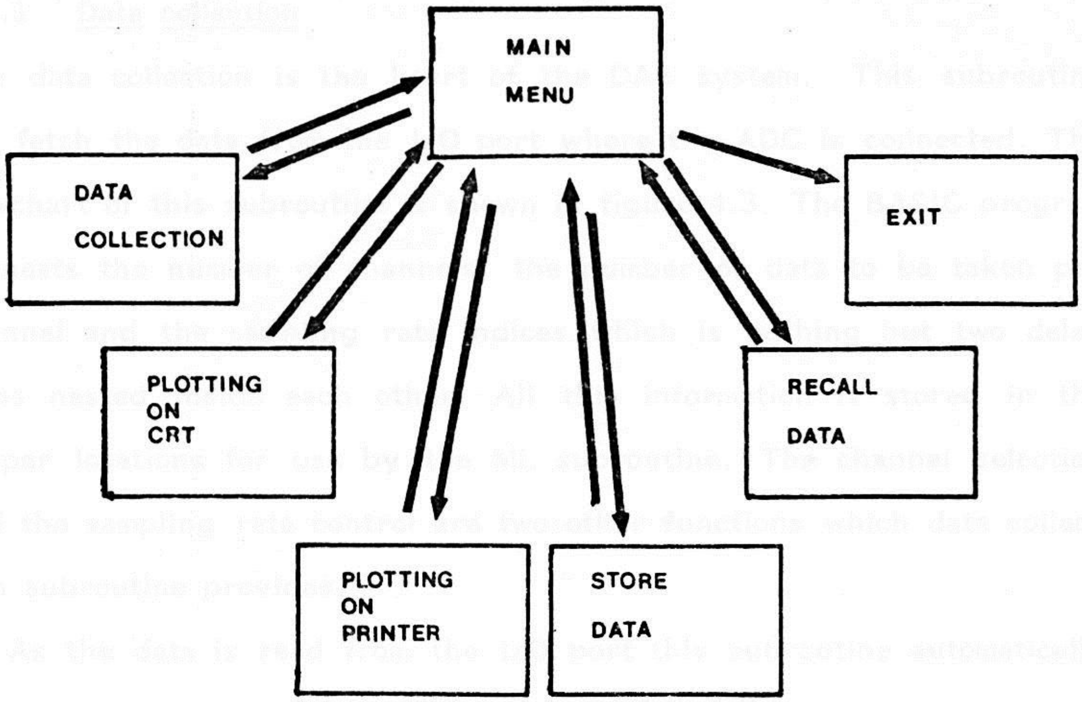


Figure 4.2: General flowchart for Data Acquisition System

4.1 Machine language subroutines

It should be mentioned that the following subroutines are not entirely ML. BASIC was used where it was necessary or where BASIC was easier to employ but the primary task is done by the ML subroutines.

4.1.1 Data collection

The data collection is the heart of the DAS system. This subroutine can fetch the data from the I/O port where the ADC is connected. The flowchart of this subroutine is shown in figure 4.3. The BASIC program requests the number of channels, the number of data to be taken per channel and the sampling rate indices which is nothing but two delay loops nested inside each other. All this information is stored in the proper locations for use by the ML subroutine. The channel selection and the sampling rate control are two other functions which data collection subroutine provides.

As the data is read from the I/O port this subroutine automatically stores the data in the 8K RAM reserved for data storage for temporary storage. Data is stored in sequential fashion. For example if three channels data are being collected the data is stored in the following order:

Location	Content
32768	X(1)
32769	Y(1)
32770	Z(1)
32771	X(2)
32772	Y(2)
32773	Z(2)
32774	X(3)
	.
	.
	.
	.
XXXXX	X(n)
YYYYY	Y(n)
ZZZZZ	Z(n)

Where X(1),X(2),X(3),...,X(n) are the first, second, third and the nth bytes of X data from channel 1. Similarlly, Y(1),Y(2),...,Y(n) are the first, second and the nth bytes of Y data from channel 2, channel 3 stores data in a like manner. There are two important points which should be realized by the programmer: the timing diagram and the interrupt system.

1. Synchronizing the software and hardware: In order to synchronize the systems software and hardware, the software had to be slowed down so the hardware (especially MUX and ADC) could keep up with the software. The timing diagram for ADC, implemented on DAS, is shown in figure 4.4. From this diagram it is necessary to introduce some delay in the software to avoid confusion and to convert the proper and valid data [8],[10].

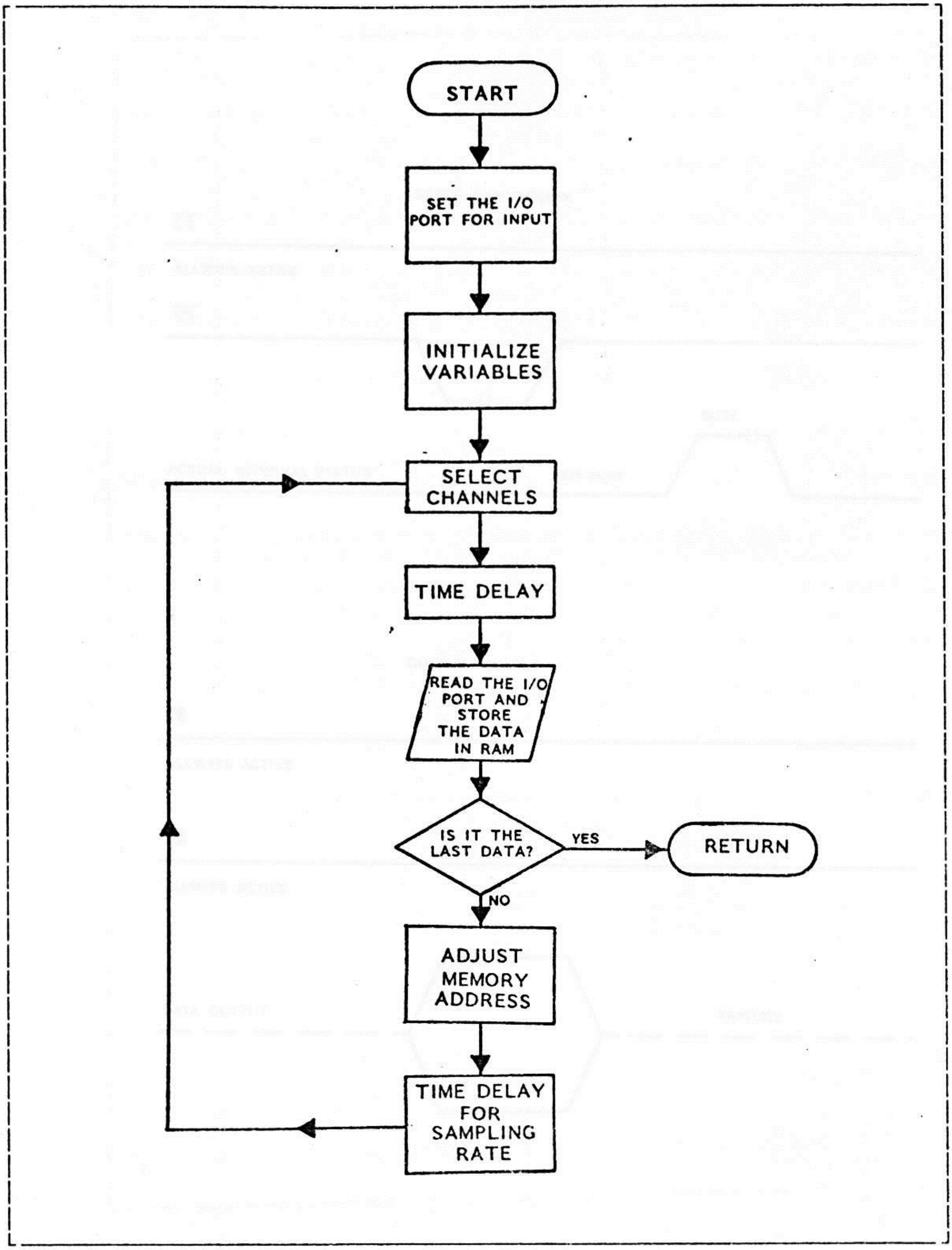


Figure 4.3: Data collection routine flowchart

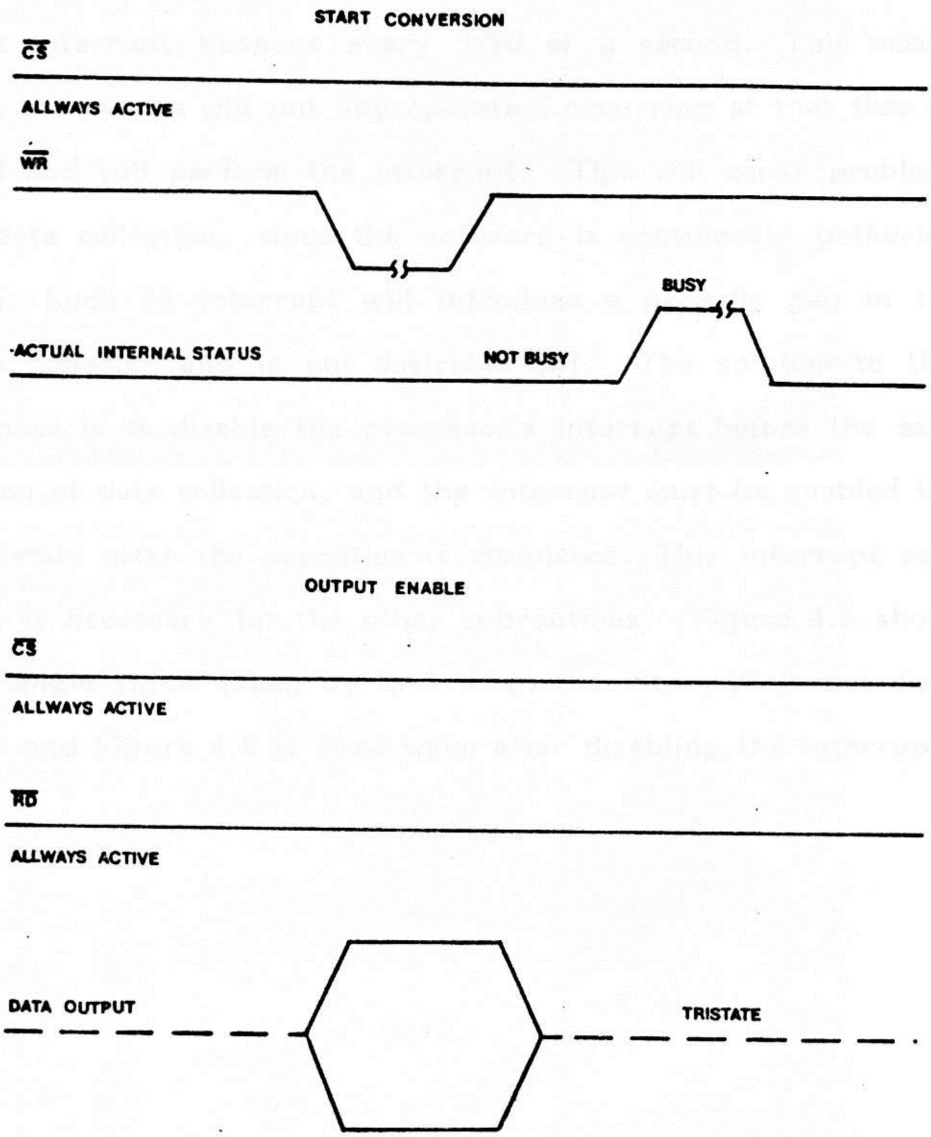


Figure 4.4: Timing diagram for Analog to Digital Converter chip

2. Interrupt service: C-64's operating system has a built-in interrupt service routine. This is used by the operating system for house-keeping tasks, i.e., scanning the keyboard for the key pressed, updating the system clock and other responsibilities. This interrupt happens every 1/60 of a second. This means that the system will put any operation occurring at that time on hold and will perform the interrupt. This will cause problems in data collection, since the software is continously gathering data. Such an interrupt will introduce a periodic gap in the data collected and is not desirable [9]. The solution to this problem is to disable the processor's interrupt before the execution of data collection, and the interrupt must be enabled immediately after the execution is completed. This interrupt service is necessary for the other subroutines. Figure 4.5 shows a triangle signal taken by DAS when the interrupt is not disabled and Figure 4.6 is same wave after disabling the interrupt.

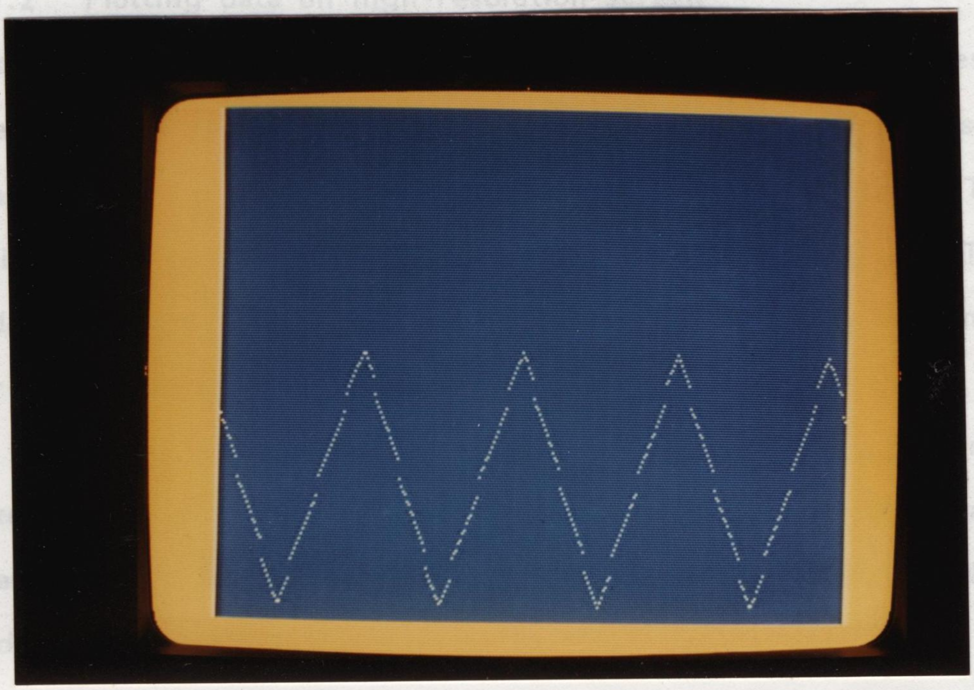


Figure 4.5: Test signal collected by DAS before interrupt disabled

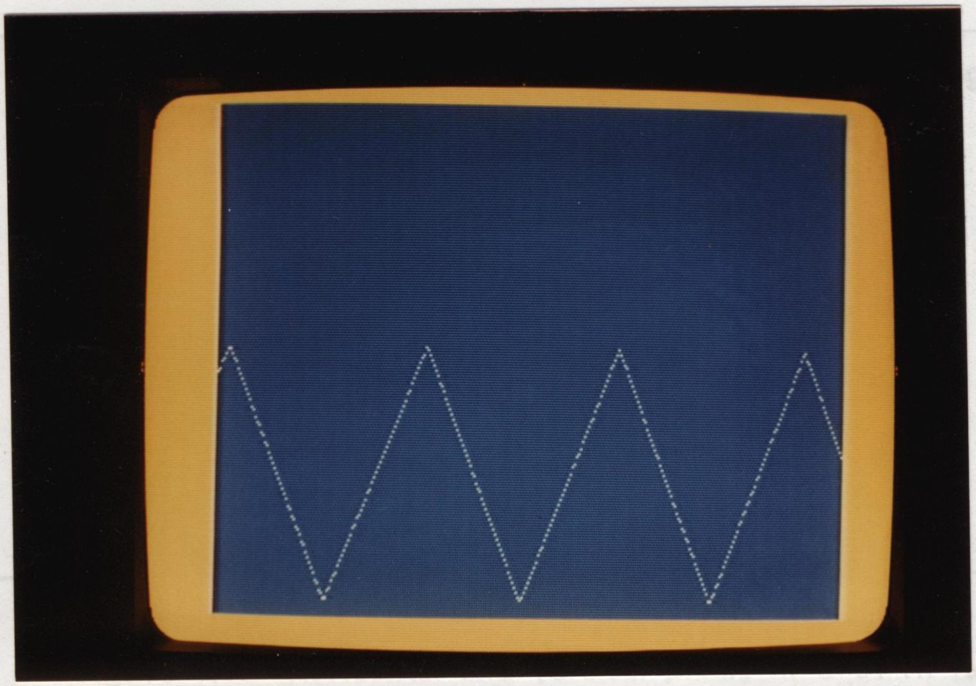


Figure 4.6: Test signal collected by DAS after interrupt disabled

4.1.2 Plotting data on high-resolution screen

After the data is collected and stored in RAM, usually it is desirable to check the validity of data by graphic representation. For this purpose the High-Resolution (Hi-Res) screen of C-64 is most suitable. The resolution of Hi-Res screen is 320 pixels (dots) by 200 pixels. This provides 64000 individual dots. This kind of resolution is sufficient enough for a adequate graphic presentation of data.

Each data point corresponds to one of the pixels; to access these pixels and turn them on or off is quite a tedious and complicated programming process (especially in ML). Figure 4.7 shows the horizontal and vertical positions of pixels.

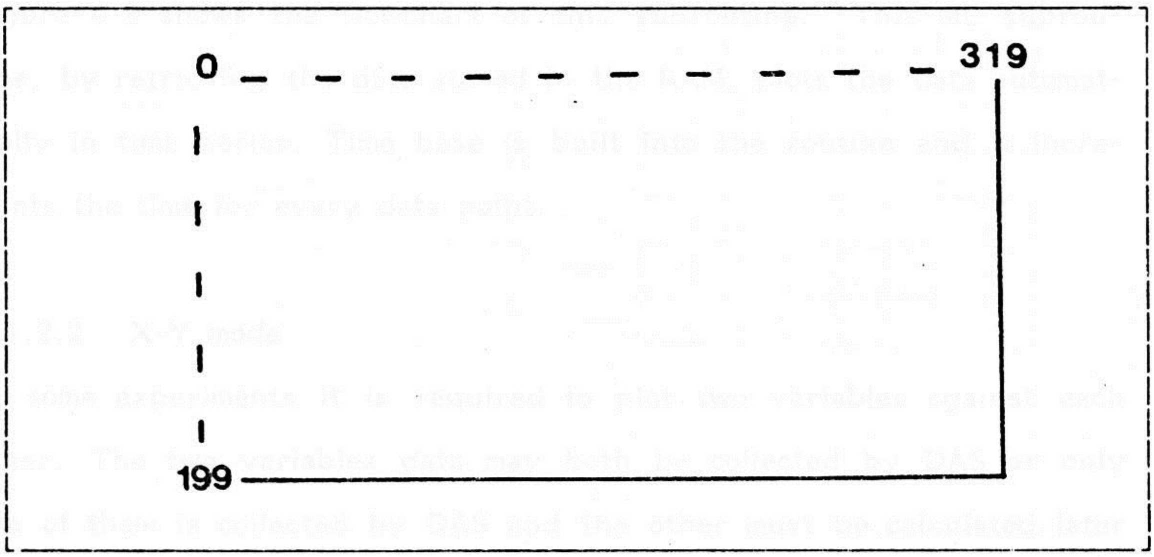


Figure 4.7: Horizontal and vertical pixels in C-64 Hi-Res

With the format given in the figure 4.7 the X and Y coordinates can be easily chosen. However, the Hi-Res screen is more complicated than

it appears. The actual arrangement of the Hi-Res screen is shown in figure 4.8.

Every byte shown in figure 4.8 consists of 8 bits or 8 pixels. Because of the awkward arrangement of bytes the access to each pixel requires further programming, which could be done in BASIC. The problem with BASIC is that it is slow, but the BASIC advantage is it's flexiblity. However, the plotting routine for the DAS system was written in ML.

Basically two different types of software for plotting were necessary: Time mode and X-Y mode.

4.1.2.1 Time mode

Figure 4.9 shows the flowchart of this subroutine. This ML subroutine, by retrieving the data stored in the RAM, plots the data automatically in time series. Time base is built into the routine and it increments the time for every data point.

4.1.2.2 X-Y mode

In some experiments it is required to plot two variables against each other. The two variables data may both be collected by DAS or only one of them is collected by DAS and the other must be calculated later by using the environments of the first variable. Thus, the two methods of X-Y mode were developed to facilitate users in both situations. Figure 4.11 shows the flowchart of this subroutine.

	★ Byte 8192	Byte 8200	Byte 8208......	Byte 8504
	8193	8201	8209	8505
	8194	8202	8210	8506
1st ROW	8195	8203	8211	8507
	8196	8204	8212	8508
	8197	8205	8213	8509
	8198	8206	8214	8510
	8199	8207	8215	8511

	Byte 8512
	8513
	8514
	8515
2ND ROW	8516
	8517
	8518
	8519

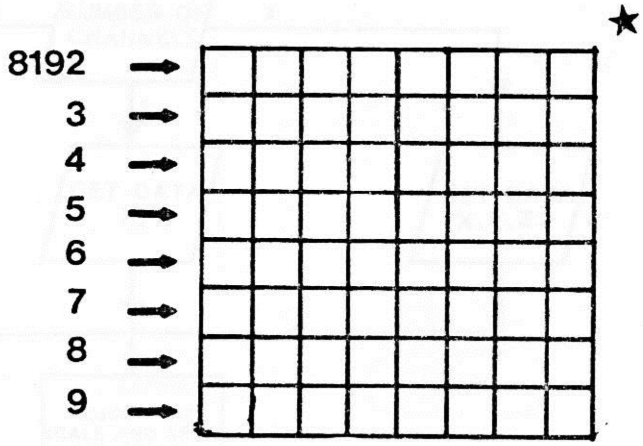


Figure 4.8: Bytes and bits arrangement of C-64's High-Resolution screen

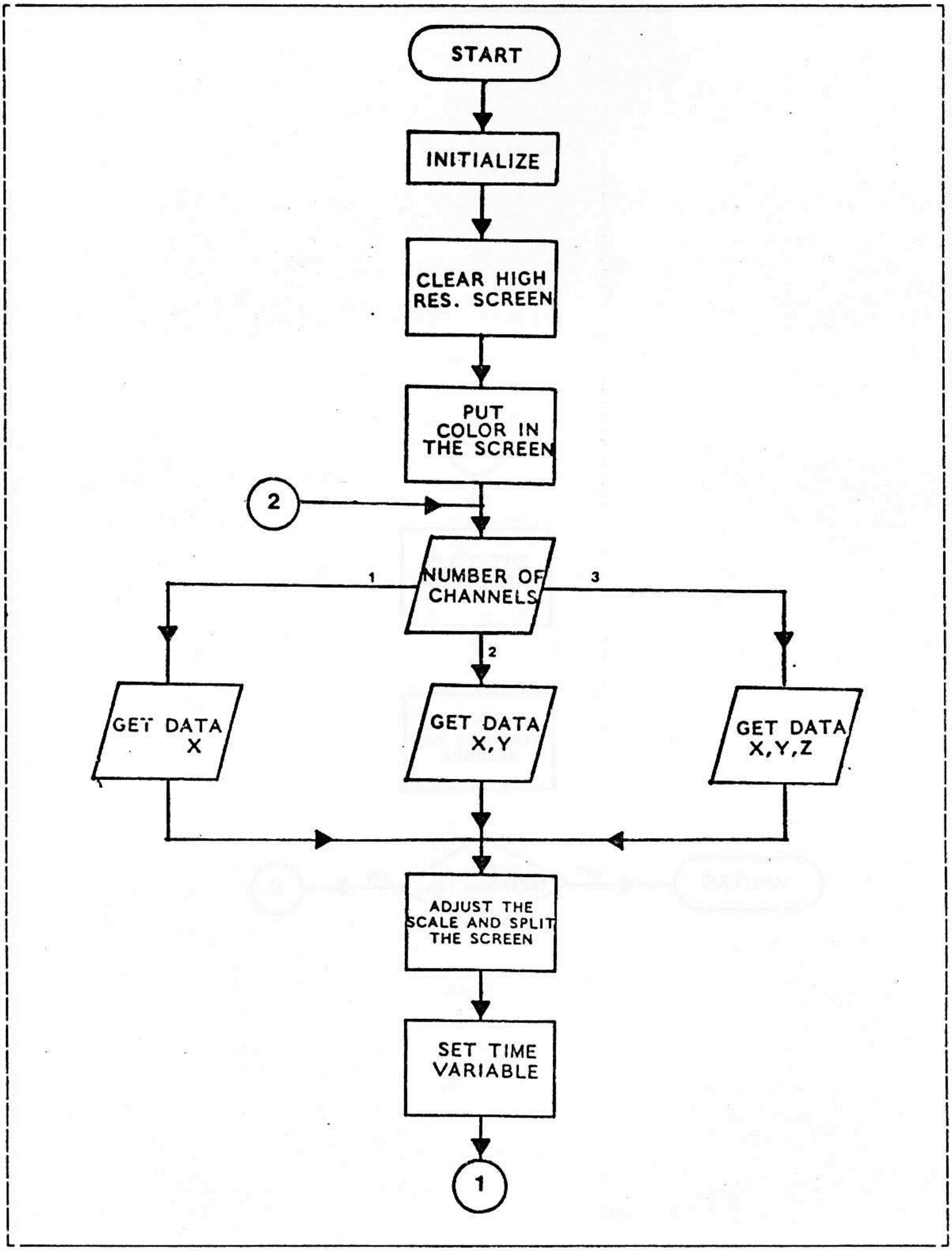


Figure 4.9: (A) Flowchart for the plotting routine (time mode)

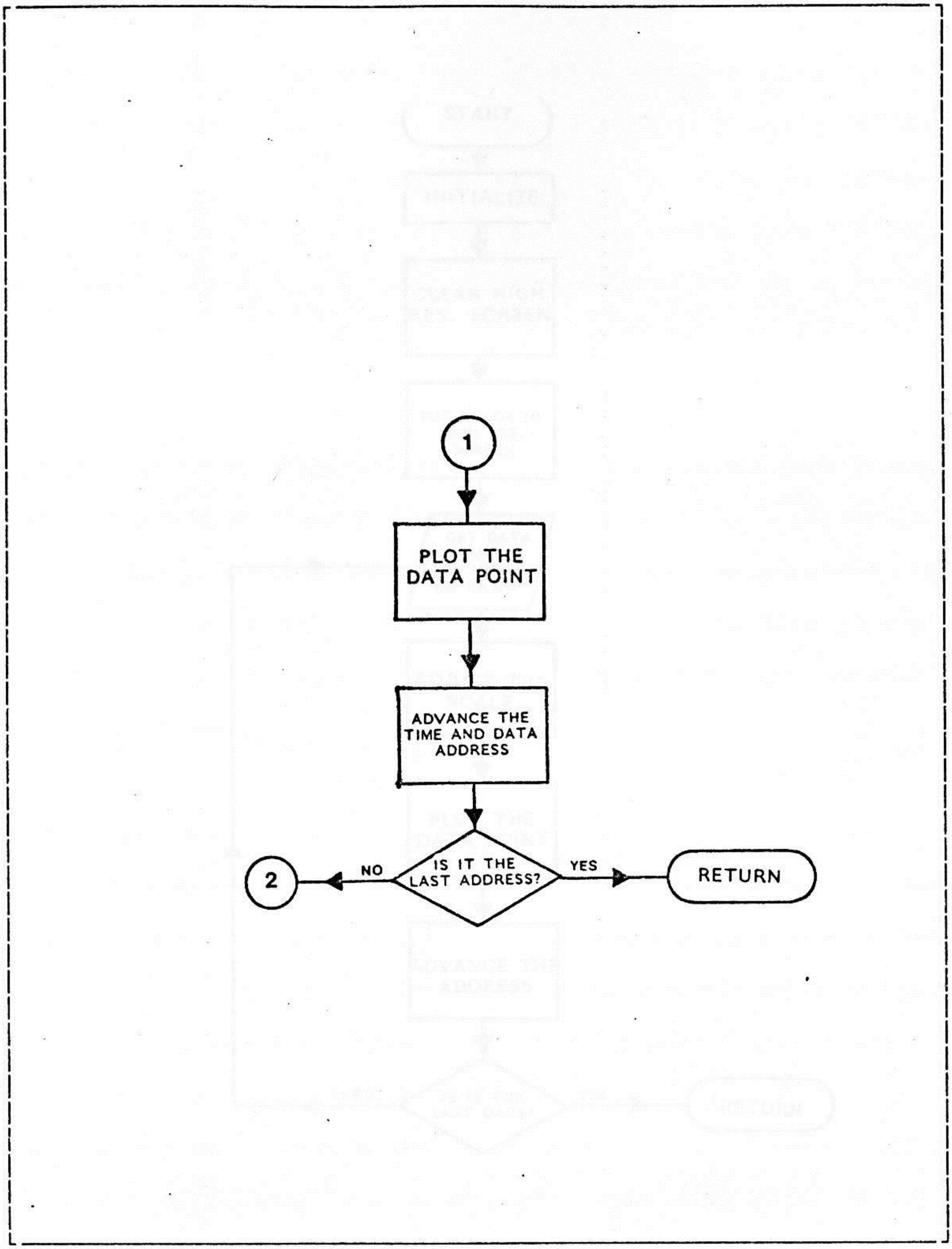


Figure 4.10: (B) Flowchart for the plotting routine(time mode)

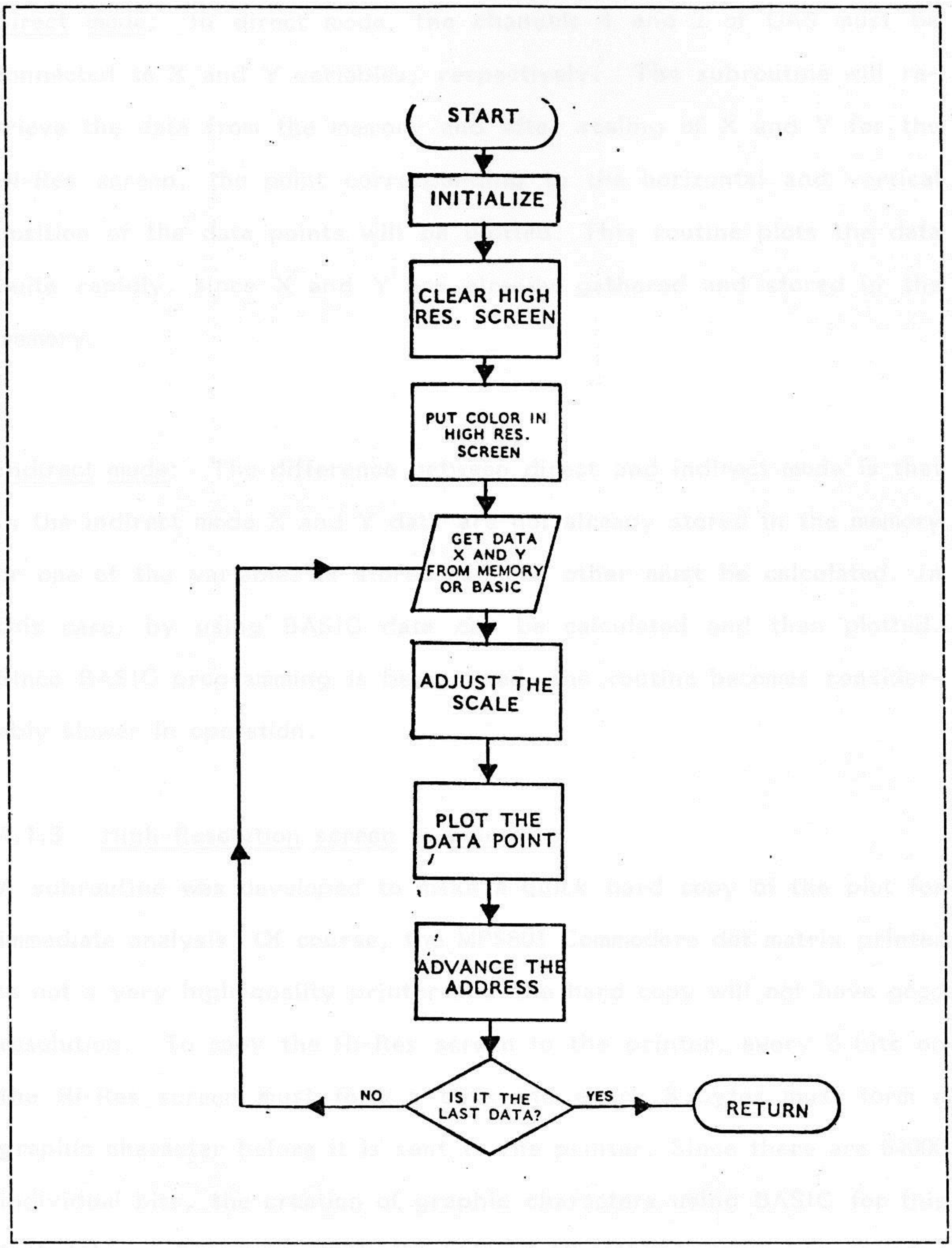


Figure 4.11: Flowchart for the plotting routine (X-Y mode)

<u>Direct mode</u>: In direct mode, the channels 1 and 2 of DAS must be connected to X and Y variables, respectively. The subroutine will retrieve the data from the memory and after scaling of X and Y for the Hi-Res screen, the point corresponding to the horizontal and vertical position of the data points will be plotted. This routine plots the data quite rapidly, since X and Y are already gathered and stored in the memory.

<u>Indirect mode</u>: The difference between direct and indirect mode is that in the indirect mode X and Y data are not already stored in the memory or one of the variables is stored and the other must be calculated. In this case, by using BASIC data can be calculated and then plotted. Since BASIC programming is being used, the routine becomes considerably slower in operation.

4.1.3 High-Resolution screen to printer

A subroutine was developed to make a quick hard copy of the plot for immediate analysis. Of course, the MPS801 Commodore dot matrix printer is not a very high quality printer and the hard copy will not have good resolution. To copy the Hi-Res screen to the printer, every 8 bits on the Hi-Res screen must form a byte and every 8 bytes must form a graphic character before it is sent to the printer. Since there are 64000 individual bits, the creation of graphic characters using BASIC for this task takes a long time. With the help of ML, this process was speeded up. Figure 4.12 shows the flowchart of this subroutine. A comparison test was made between the BASIC and ML execution time. The BASIC

routine took about 8 minutes while ML took less than 2 minutes. The complete listing of this subroutine is in appendix E.

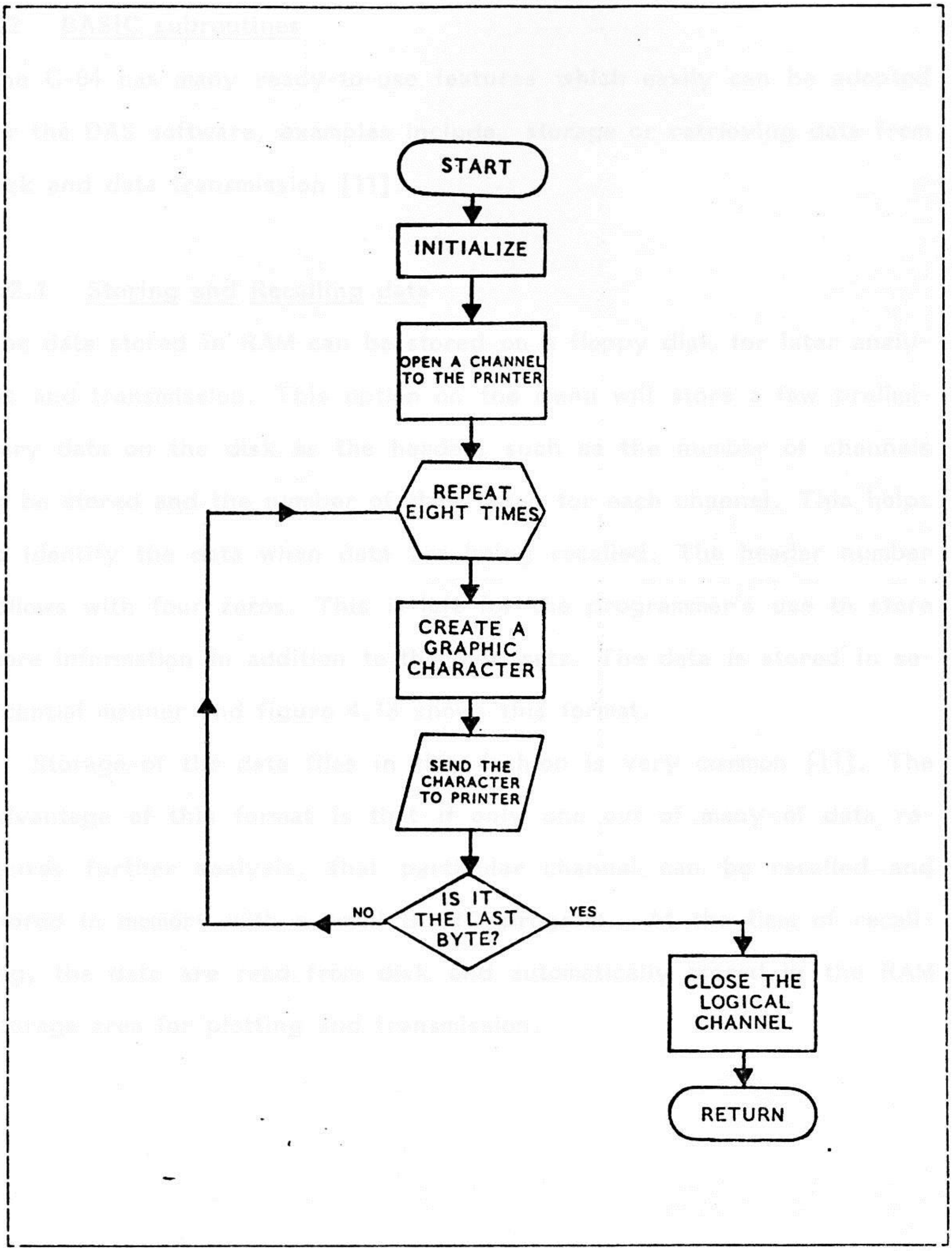


Figure 4.12: Flowchart of the printer routine

4.2 BASIC subroutines

The C-64 has many ready-to-use features which easily can be adopted for the DAS software, examples include, storage or retrieving data from disk and data transmission [11].

4.2.1 Storing and Recalling data

The data stored in RAM can be stored on a floppy disk for later analysis and transmission. This option on the menu will store a few preliminary data on the disk as the header, such as the number of channels to be stored and the number of data taken for each channel. This helps to identify the data when data are being recalled. The header number follows with four zeros. This is left for the programmer's use to store more information in addition to the data sets. The data is stored in sequential manner and figure 4.13 shows this format.

Storage of the data files in this fashion is very common [11]. The advantage of this format is that if only one out of many of data requires further analysis, that particular channel can be recalled and stored in memory with a small BASIC program. At the time of recalling, the data are read from disk and automatically stored in the RAM storage area for plotting and transmission.

```
3          Number of channels;
320        number of data per each channels;
0
0          Header of data
0
0
X(1)       first data
Y(1)       second data
Z(1)       third data
 .
 .
 .
X(n)
Y(n)
Z(n)
```

Figure 4.13: Format of data storage on the disk

4.2.2 Transmission

The data stored in RAM can be transmitted through co-axial cable from laboratories to a TRS-80 computer in form of serial transmission [7],[9]. The C-64 has a built-in RS-232 for this type of transmission. This will be explained in greater detail in chapter 6.

Chapter V
THE EXPERIMENTS

One of the main objectives of this thesis is to computerize the existing instrumentation in the Mechanical engineering laboratories. The purposes of this chapter are:

1. To demonstarte the effectiveness of DAS;
2. A general comparison of old methods (analog) to new computerized methods (digital);
3. To show the extensive data base analysis capability of the digital method compared to the time-consuming analog method;
4. To change the laboratories experimental procedures so that they are compatible with current computer base technology and readily available in the undergraduate curriculum as suggested by ABET;

Although DAS can be employed in almost every phase of mechanical engineering, two experiments which are included in the mechanical engineering curriculum are presented in this chapter. The examples demonstrate the effectiveness and versatility of DAS in mechanical engineering education and will prepare students in the modern instrumentation technology which they will inevitably encounter after graduation. A third example is an experiment in an actual industrial application.

5.1 Cam analysis experiment

The objective of this experiment was to find the motion of a cam for educational and machine design purposes. Study of displacement, velocity and acceleration of a cam is important. As for the synthesis of cam analysis the displacement must be known to satisfy motion requirements. The cam velocity and acceleration also should be evaluated to analyze the stress applied on the cam to avoid rapid wear and breakage. In the Mechanical Engineering Measurement Laboratory there are three transducers connected to the cam apparatus: displacement, velocity and acceleration. However in this experiment only the velocity transducer was used because the acceleration transducer is an undamped type and displacement transducer at the time of operation produces 500 Hz carrier frequency which appears as noise. The method employed in this thesis to get all three quantities (displacement, velocity and acceleration) was to use the velocity transducer and by taking the integral and derivative of the velocity signal to produce the displacement and acceleration, respectively.

The output voltage of the velocity transducer takes a positive or negative value according to the direction of movement of the object of interest. The ADC as designed in this thesis project can only accept voltages between 0V and +5V. Therefore, the output of the velocity transducer must be conditioned so that the signal falls in the 0-5V range. Some amplification is neccessary to achieve maximum resolution. The output of the velocity transducer is in the order of volts. An analog computer can be used as a signal conditioning device since the operational amplifiers with various resistors and capacitors are available.

5.1.1 calibration of velocity transducer

The displacement of the cam was directly measured by a dial gauge (mechanical means) and the calibration of amplifiers was verified. The output of the velocity transducer was connected to an analog computer and DAS. Figure 5.1 shows this arrangement. The motor was started and velocity data was taken at the rate of 1000 samples per second. Figure 5.2 shows the plot of this velocity data.

Next, by using an integration subroutine written in C-64 BASIC the area under the first part of the velocity curve (indicated on figure by 1-2-3) was determined. Using the manufacturer's sensitivity factor, the result should correspond to the maximum displacement measured by the dial gauge, which was 0.16 inches. The cross calibration of these values did not agree with each other. Table 5.1 shows the result of this cross calibration.

The only conceivable cause of error was the velocity transducer sensitivity factor. The velocity transducer consists of a magnetic core,coil and housing. Figure 5.3 shows the velocity transducer equivalent circuit. The sensitivity factor is directly dependent on the magnetic core of the transducer.

$$e=B\ L\ V \qquad \text{eq. 5.1}$$

Where e=output of transducer (Volts)

B=magnetic filed of the core (const.)

L=the length of magnetic core (const.)

V=velocity

The B and L are constant, so:

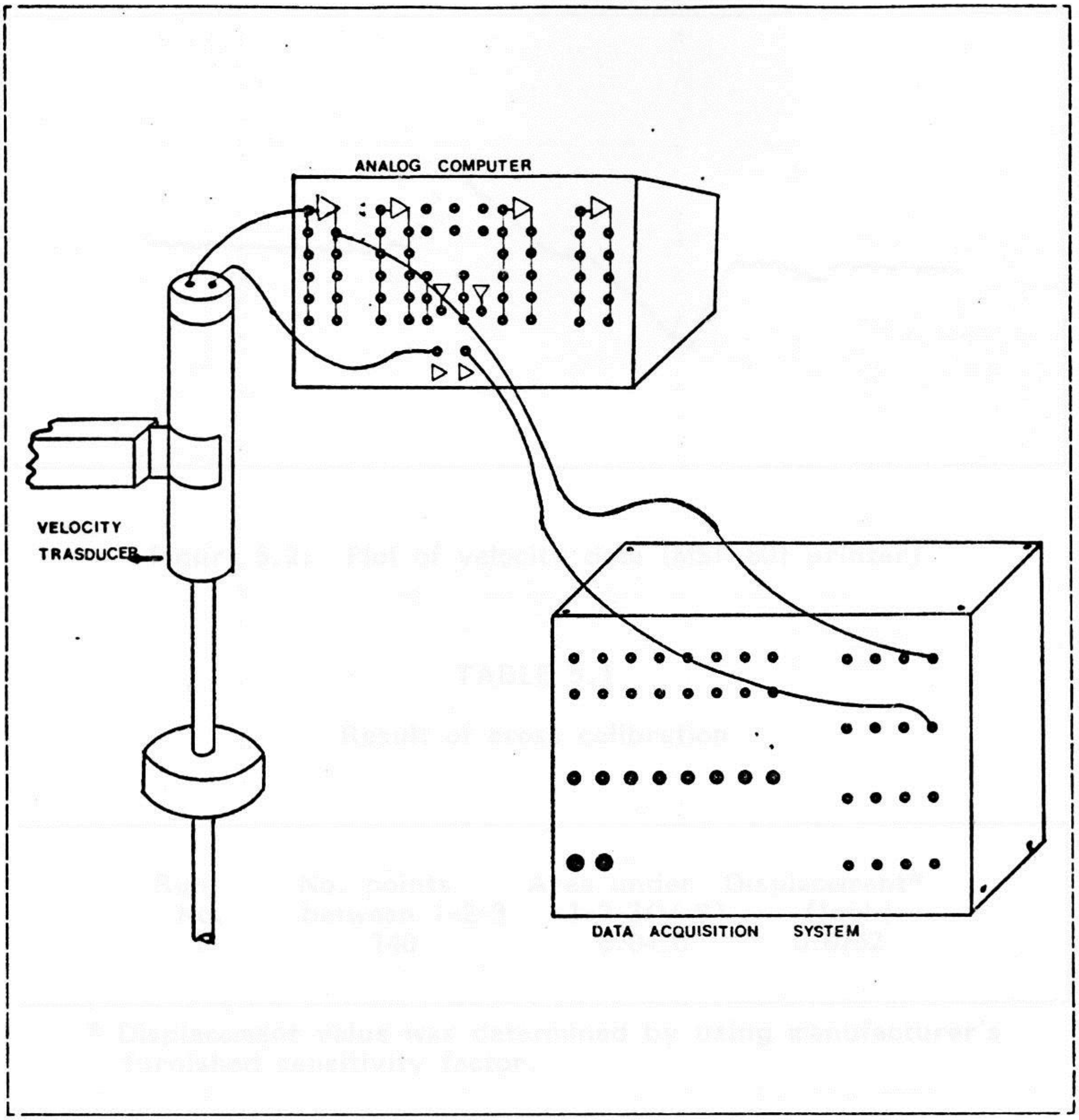


Figure 5.1: Connection of DAS for velocity transducer calibration

$$e \propto V \qquad \text{eq. 5.2}$$

The demagnetization of the magnetic core will change the sensitivity factor of the transducer [12]. Thus, further investigation and calibration is neccessary.

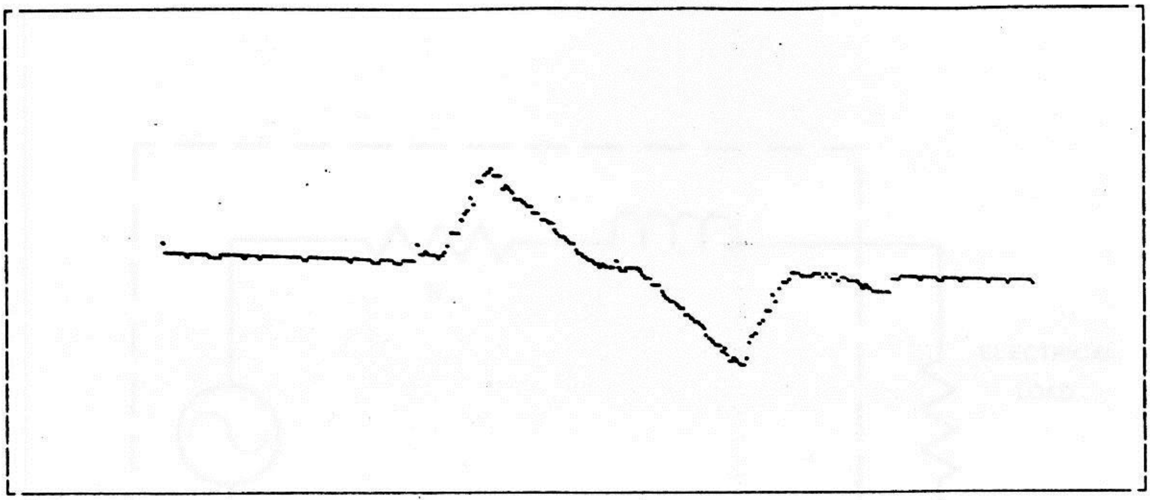


Figure 5.2: Plot of velocity data (MSP-801 printer)

TABLE 5.1

Result of cross calibration

Run No.	No. points between 1-2-3	Area under 1-2-3(V·S)	Displacement* (Inch)
1	140	0.0420	0.0752

* Displacement value was determined by using manufacturer's furnished sensitivity factor.

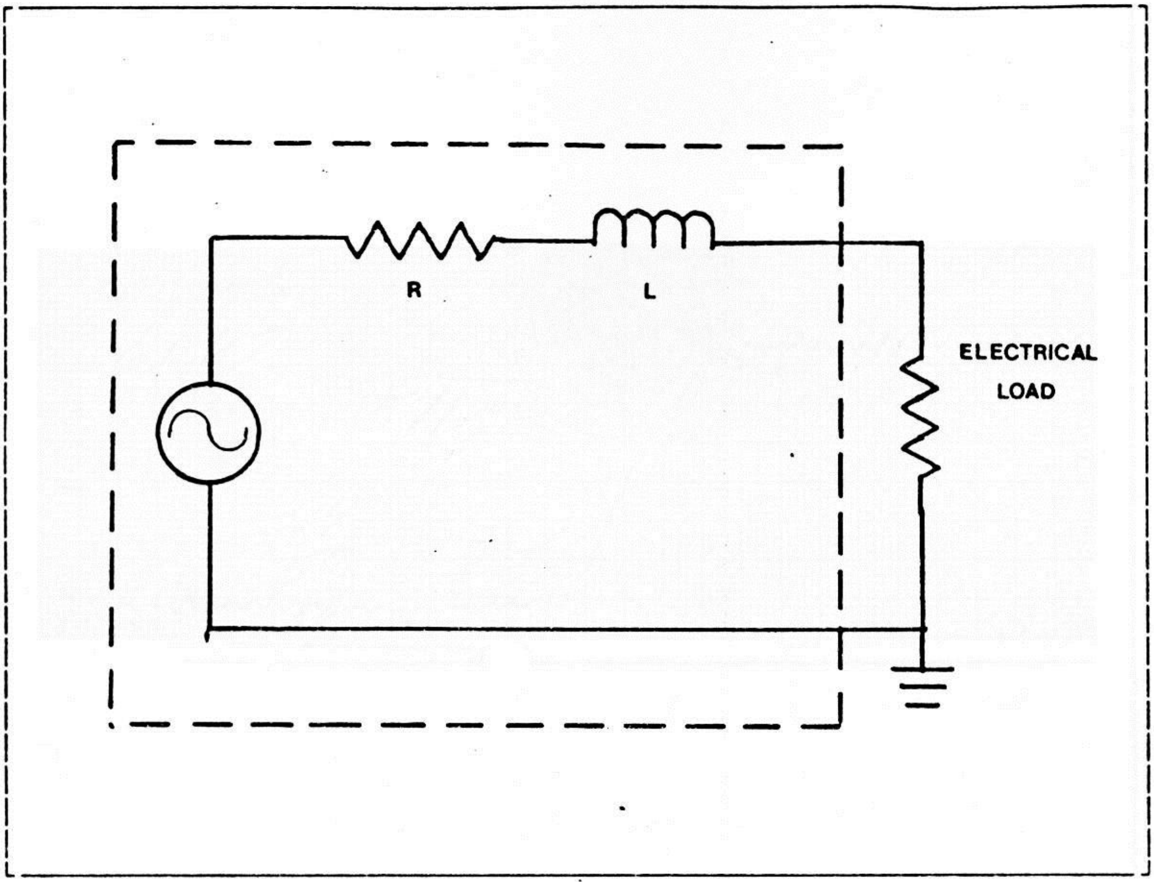


Figure 5.3: Velocity transducer equivalent circuit

5.1.2 method of calibration

The method explained here has been developed by the author and K. Okamura. It appears to be more direct and reliable than the one used by the manufacturer. Figure 5.4 shows the manufacturer's calibration curve furnished with the velocity transducer.

To apply this new method several assumptions were made:

1. The resistance of magnetic field and other resistances are negligible.
2. The velocity transducer is a linear device, from eq.5.2.

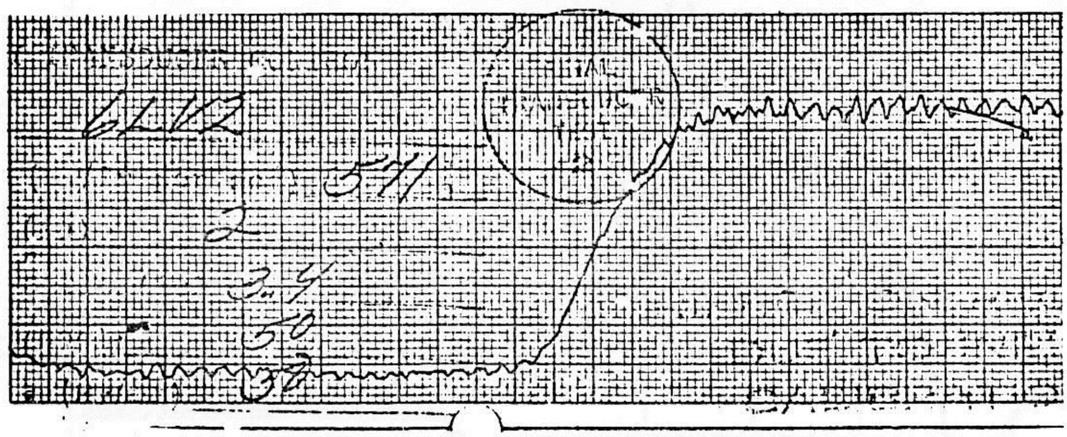


Figure 5.4: Manufacturer calibration curve for velocity transducer

The output of the transducer was connected to the analog computer and DAS as shown on figure 5.1. The magnetic core was manually lifted at some known height and released. A cushion was placed under the core to lessen the impact force on a hard surface and to avoid further demagnitization. As the core was released the output was recorded by DAS. A plot of this output is shown in figure 5.5.

The slope of the first linear portion of this output is of interest. The ratio of the slope and the gravitational acceleration (g=32.2 ft/sec/sec) is the sensitivity factor. The theory behind this method is ex-

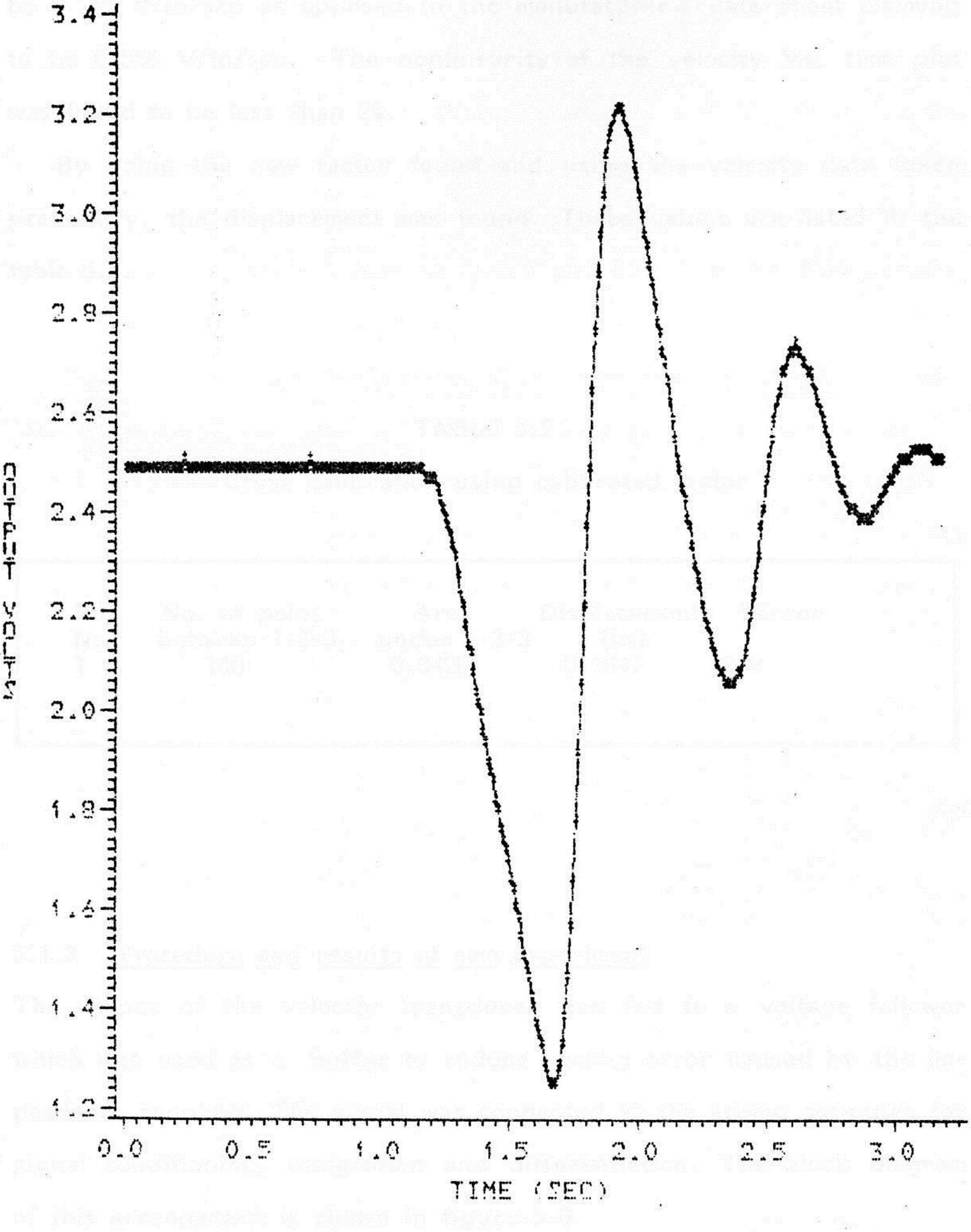


Figure 5.5: Velocity output for calibration

plained in appendix A. From this the sensitivity factor was found to be 0.255 V/in/sec as opposed to the manufaturer's data sheet claiming to be 0.558 V/in/sec. The nonlinearity of the velocity Vs. time plot was found to be less than 2%.

By using the new factor found and using the velocity data taken previously, the displacement was found. These values are listed in the table 5.2.

TABLE 5.2

Cross calibration using calibrated factor

Run No.	No. of point between 1-2-3	Area under 1-2-3	Displacement (in)	%Error
1	140	0.0420	0.1647	2.9

5.1.3 Procedure and results of cam experiment

The output of the velocity transducer was fed to a voltage follower which was used as a buffer to reduce loading error caused by the impeadance coupling. The signal was connected to the analog computer for signal conditioning, integration and differentiation. The block diagram of this arrangement is shown in figure 5.6.

The output of the integrator, amplifier and differentiator were fed to the buffers and from there to channels 1, 2 and 3 of DAS, respectively. The motor was started and the RPM was adjusted and measured. Then the data was taken by C-64 and stored on a disk. After the experimental session the collected data was transmitted to the main frame computer and by using the appropriate factors the data was plotted to the appropriate units. Figures 5.7, 5.8 and 5.9 show the displacement, velocity and acceleration plots of the cam.

Figure 5.10 shows the profile of the cam used in the experiment. The experiment was done as a reverse engineering method for educational purposes, by converting the time to degrees and using the displacement data for one complete cycle. Then by choosing some arbitrary radii for the prime-circle and follower and with use of SAS[4] graphics, the profile of the cam was graphically determined [13],[14].

---

[4] Statistical Analysis System package available at NDSU.

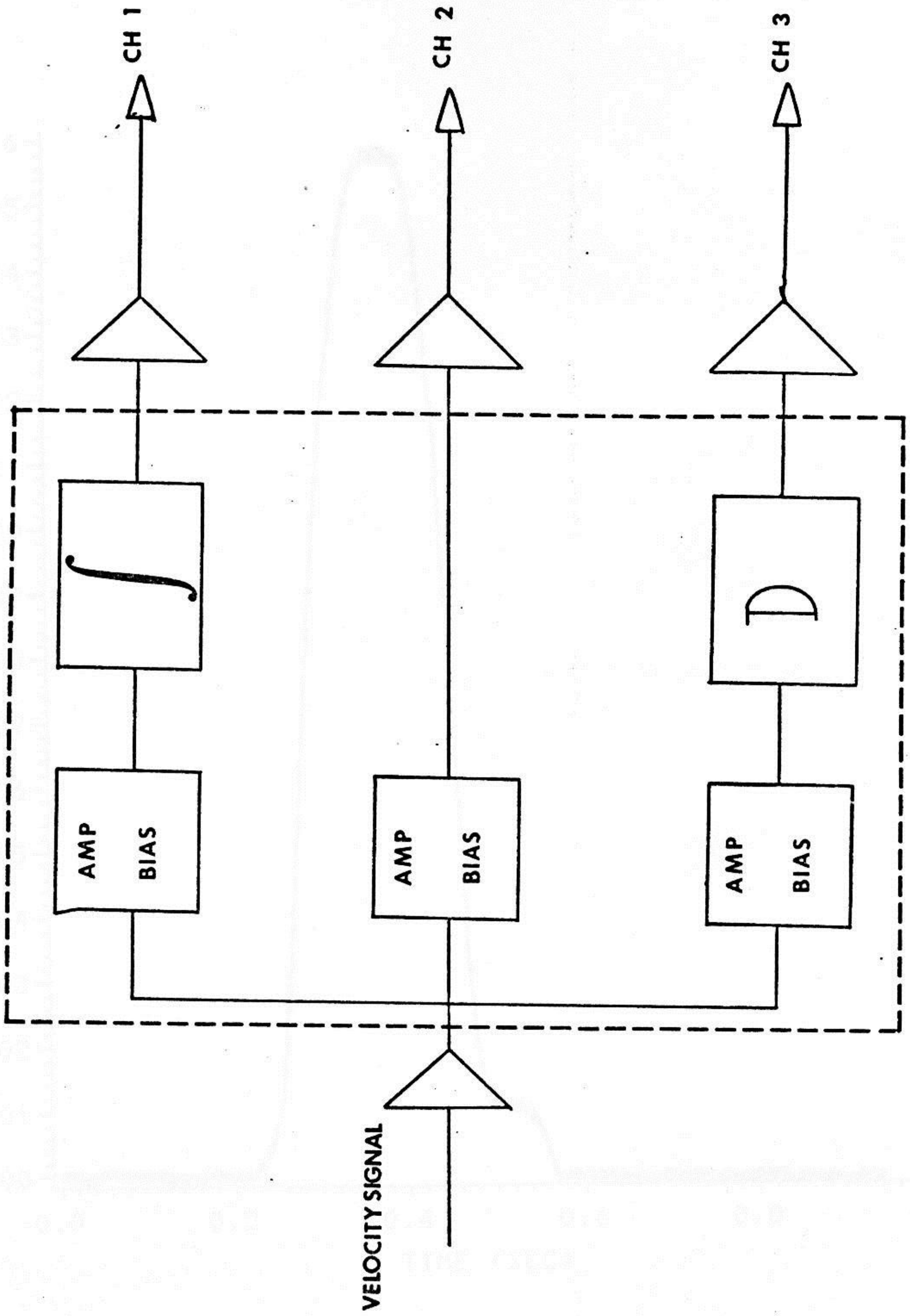


Figure 5.6: Block diagram for cam experiment

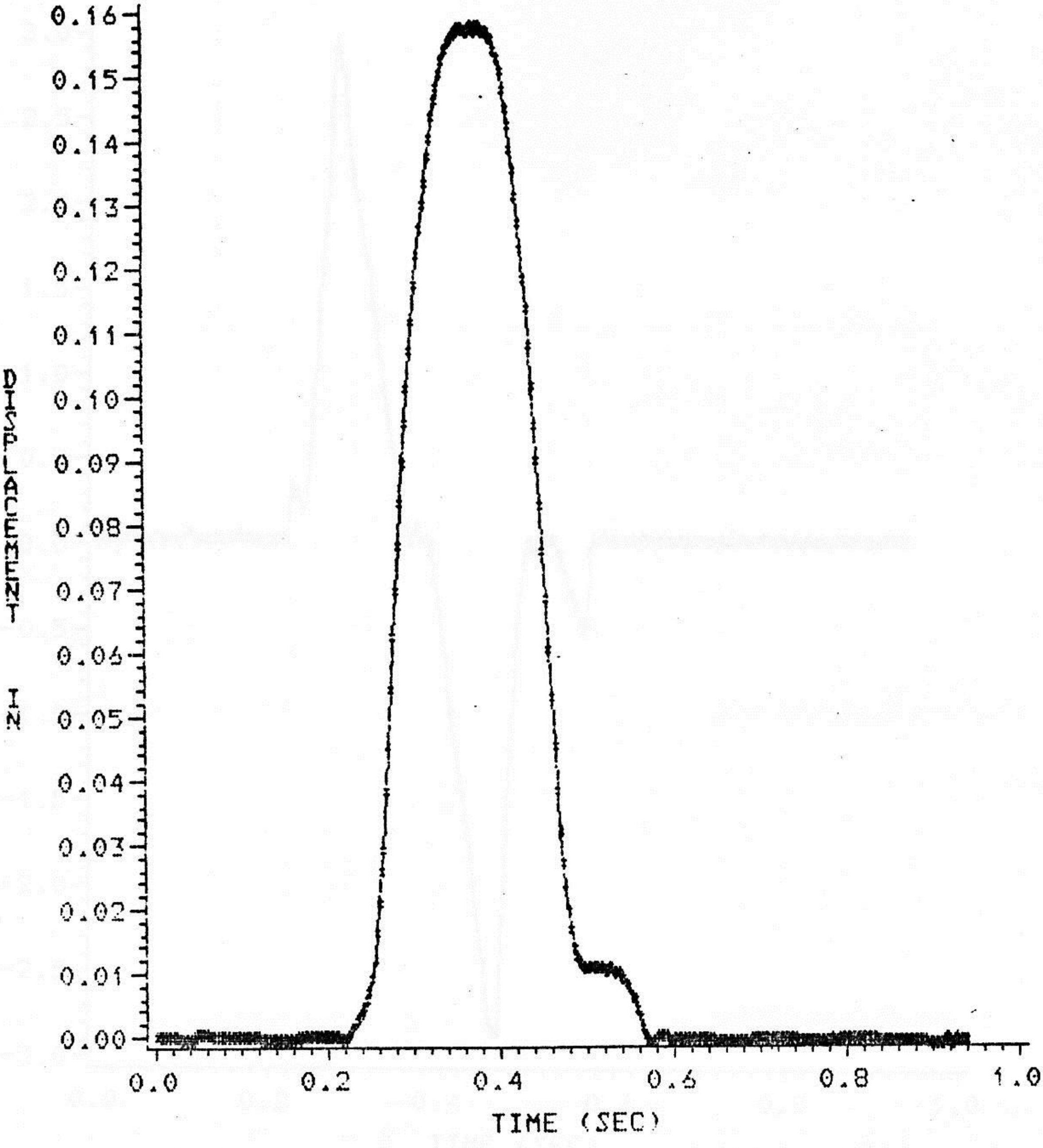


Figure 5.7: Displacement of the cam

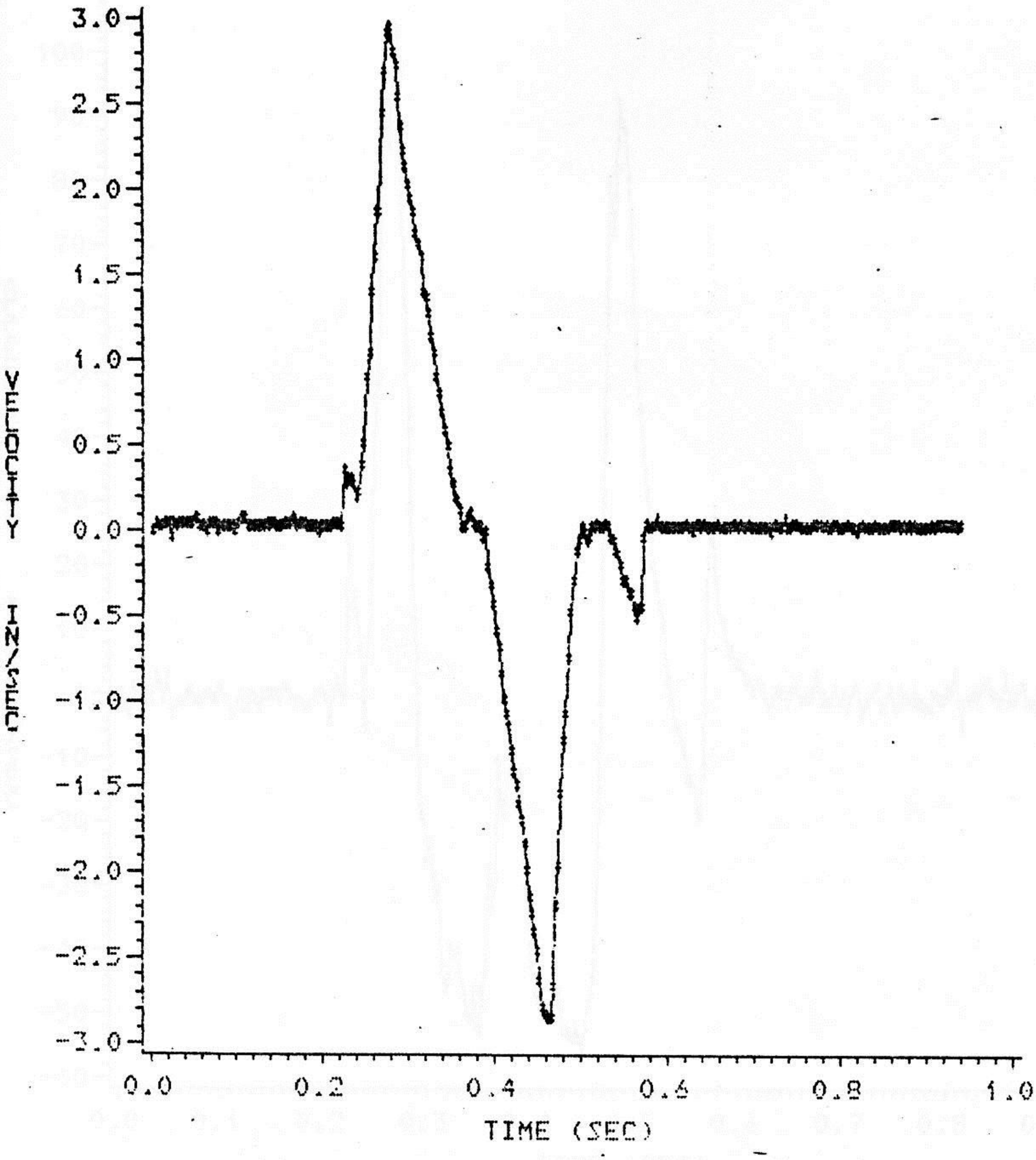


Figure 5.8: Velocity of the cam

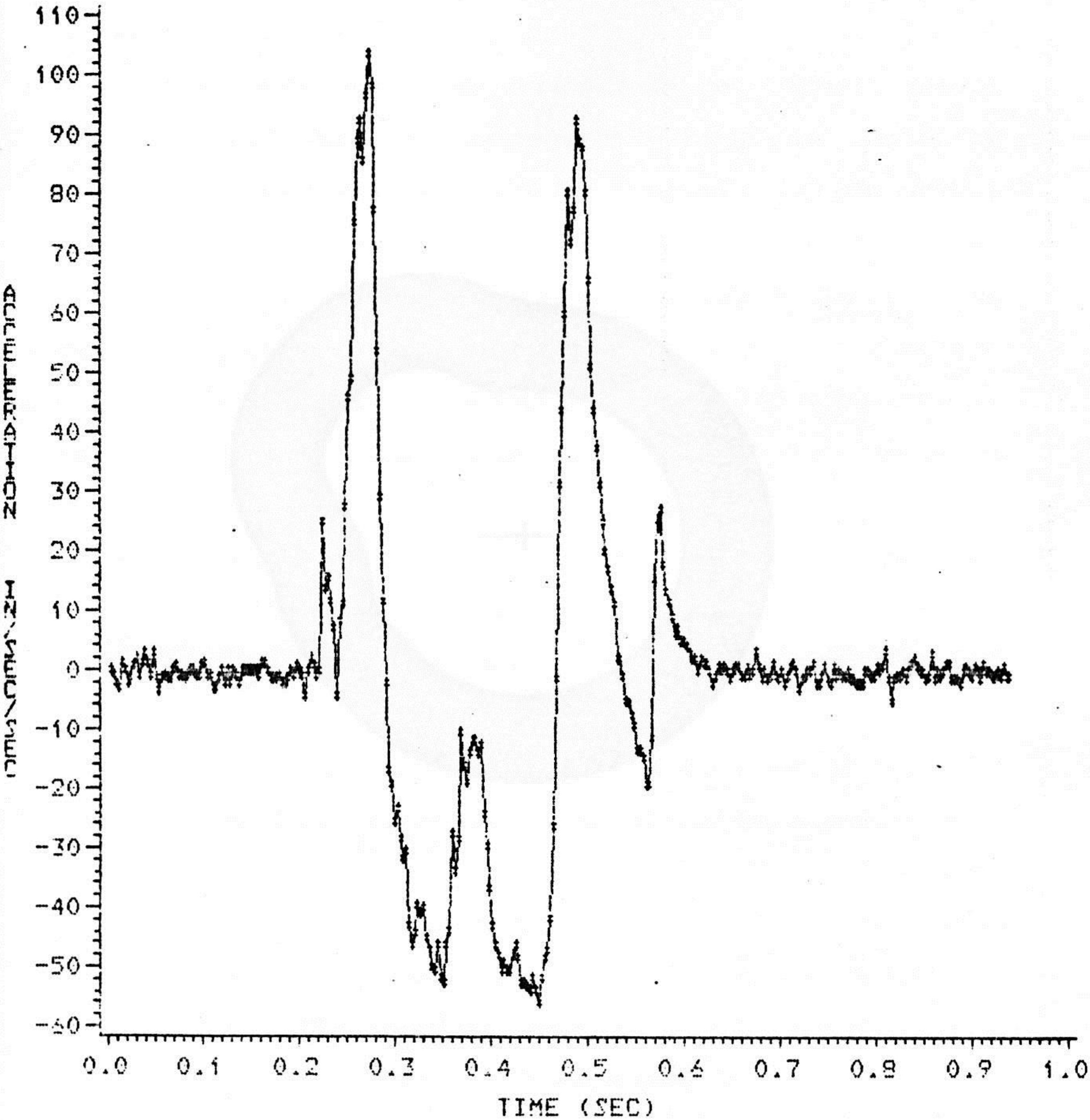


Figure 5.9: Acceleration of the cam

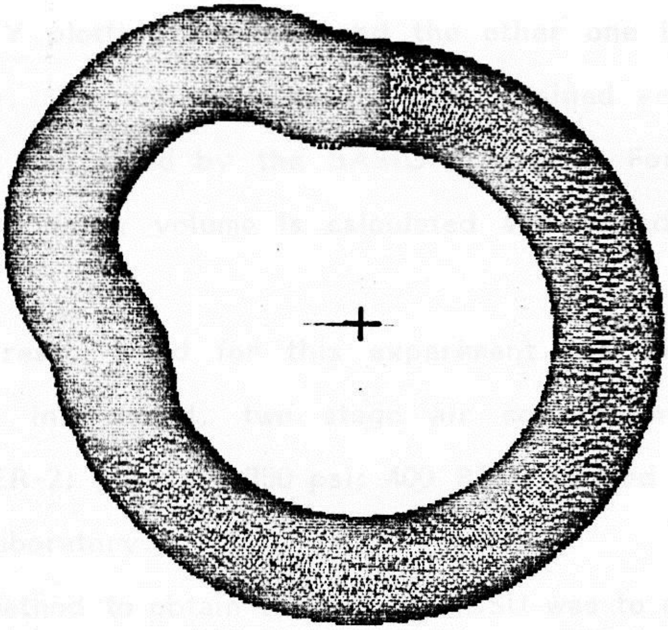


Figure 5.10: The cam profile

5.2 Air compressor experiment

The purpose of this experiment was to attain a convenient, accurate and fast Cylinder Performance Indicator diagram (CPI) of the compression process.

Customized software has been developed for the compressor experiment (see appendix B). This new software is basically the same as the general data acquisition software with the exception of two new options: one is display of a Pressure-Volume (PV) diagram by use of the indirect mode X-Y plotting routine, and the other one is display of the disk directory. To plot the pressure data obtained versus the volume, the volume is calculated by the BASIC program. For every pressure data a corresponding volume is calculated and immediately plotted on CRT.

The compressor used for this experiment was the Ingersoll-Rand single-acting, intercooled, two stage air compressor (serial number 75118; class ER-2; pressure=350 psi; 400 RPM) located in the mechanical engineering laboratory.

The old method to obtain the CPI at NDSU was to use an engine indicator device which was positioned near a rotating cylinder covered with paper. When the force in the cylinder equals the pressure acting on the spring of engine indicator the pen moves and makes a mark on the paper. At the same time string is connected to the crankshaft of the compressor. This moves the paper and indicates the displacement of the cylinder. The displacement is reduced to fit on the small paper on the engine indicator, thus, creating a plot representing the PV diagram.

Figure 5.11 shows the engine indicator which has been used for many years in the Mechanical Engineering Laboratory. The CPI created by the engine indicator is analog data in graphic form. For most performance evaluations many calculations and numerical analysis must be done. Thus, one must take data points from the small plots by relying on visual accuracy. This induces more errors in the calculations.

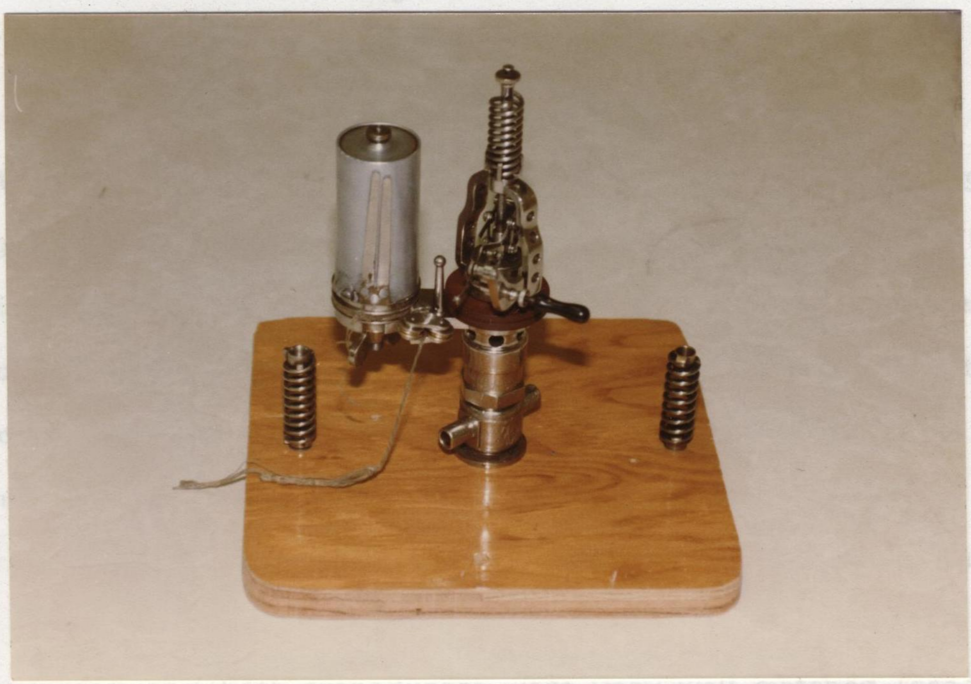


Figure 5.11: Engine indicator used to plot CPI

By employing DAS, many data can be taken and stored in form of digital values, thus the numerical analysis and calculations are much

easier, more accurate and also the data can be manipulated in many more ways. The data can be plotted in many different sizes and styles to fit a particular purpose.

Two pressure transducers were installed to monitor the low and high pressure cylinders. A photo transistor and an infrared emitter source were installed on the compressor flywheel. This photo transistor acts as an indicator for the start and end of a cycle. The pressure transducer and amplifier needed to be calibrated and the calculation method for volume needed further investigation.

5.2.1 Calibration of pressure transducers

The transducers used in this experiment were Statham strain-gage-diaphragm (model No.: PG 3288 TC) type pressure transducers.[5] The fast reaction time and the output range made this type of transducer very attractive for this experiment. Calibration was neccessary since the excitation voltage used for the transducer was five volts while the manufacturer tested and calibrated using 10V.

A dead-weight tester was used to calibrate the pressure transducers. The result of the calibration shows extreme linearity, with almost no hysteresis and both transducers were very well matched (see appendix G for calibration data).

---

[5] The manufacturer's data sheet is included in appendix G.

5.2.2 Calibration of the amplifiers

The AD522 amplifiers (the circuitry explained in chapter 3) were calibrated by creating millivolts from a voltage divider circuit as input to the amplifiers. The input and output of the amplifiers were monitored by HP-3465A multimeter[6] which has a micro volt resolution and accuracy. This was done for all three pre-selected gains of 1000, 500 and 100 for both amplifiers. The result of calibration (included in appendix F) shows that the amplifiers are highly linear and have very low drift.

5.2.3 Volume calculation

In the data acquisition, analog or digital, a P-V diagram is determined with time as an implicit parameter. In the case of DAS, the photo sensor output indicates the time marker representing the top dead center of the low pressure cylinder. This is shown on figure 5.12. The top plot is the low pressure against time, the middle is high pressure against time and the bottom plot is the time indicator. Figure 5.13 and 5.14 shows the circuit, place and position of the marker in an actual system. The relationship between the angular velocity and angle θ is:

$$\theta = \omega t \qquad \text{eq. 5.3}$$

This relationship holds true if the angular velocity (in other words, the RPM of compressor) is constant. To verify this, four equal size blocks were made and installed on the flywheel of the compressor at every 90 degrees. The compressor was started and the output of the photo transistor was taken via channel 1 of DAS.

---

[6] Hewlett Packard company.

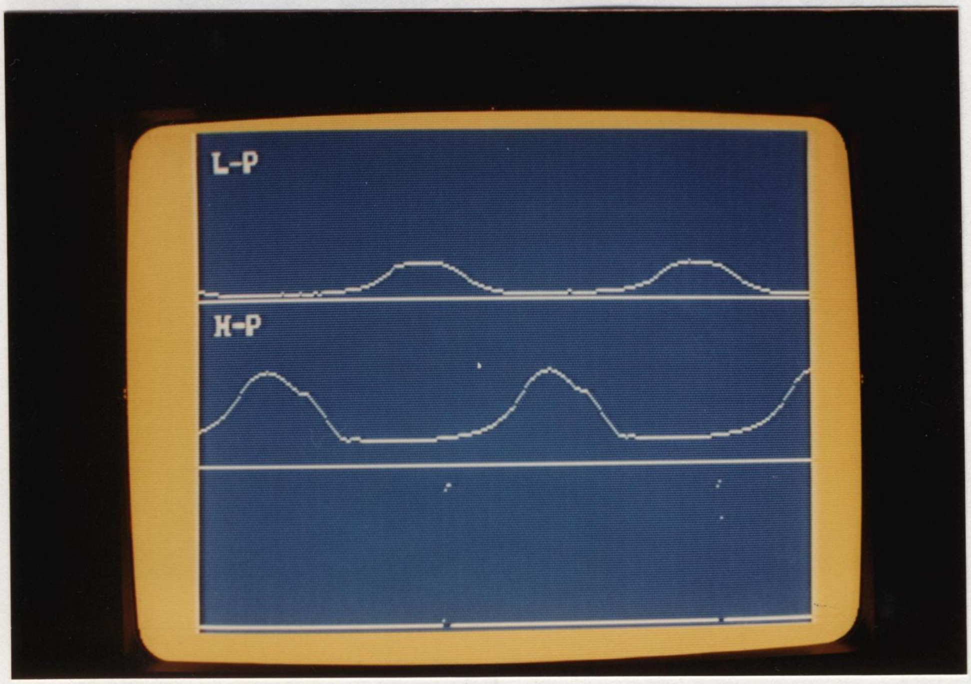


Figure 5.12: Photo 1 low and high pressure data and time marker

The number of data points between each marker (quarter) were counted to see if the rotation of the crankshaft is at a constant speed. The result of this experiment is shown in table 5.3.

From the result tabulated in table 5.3 the constant RPM is verified. Since the RPM is constant, the time t can represent the angle θ.

$$\theta \propto t \qquad \text{eq. 5.4}$$

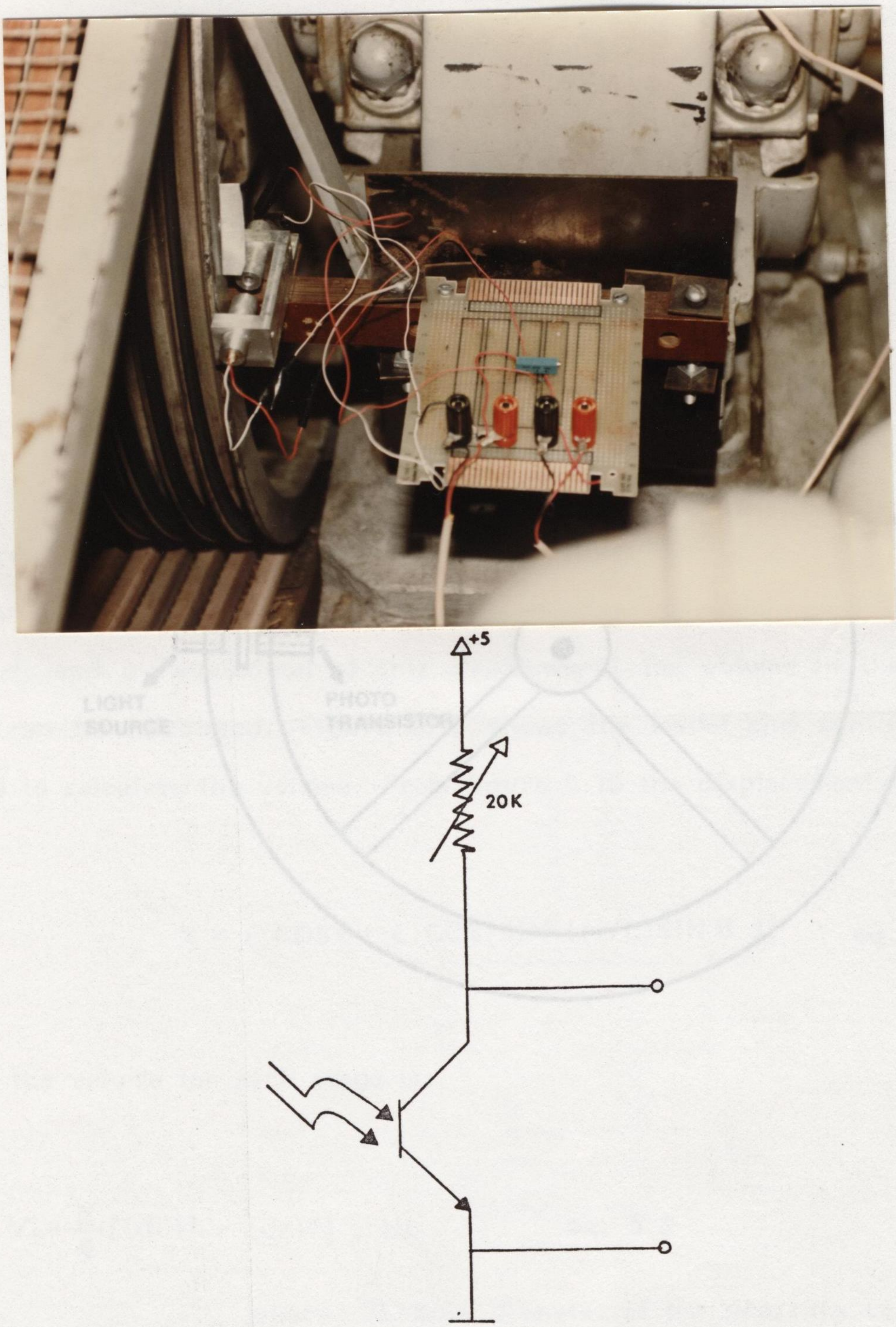


Figure 5.13: (A) Place and Position of photo transistor (B) Circuit diagram for photo transistor

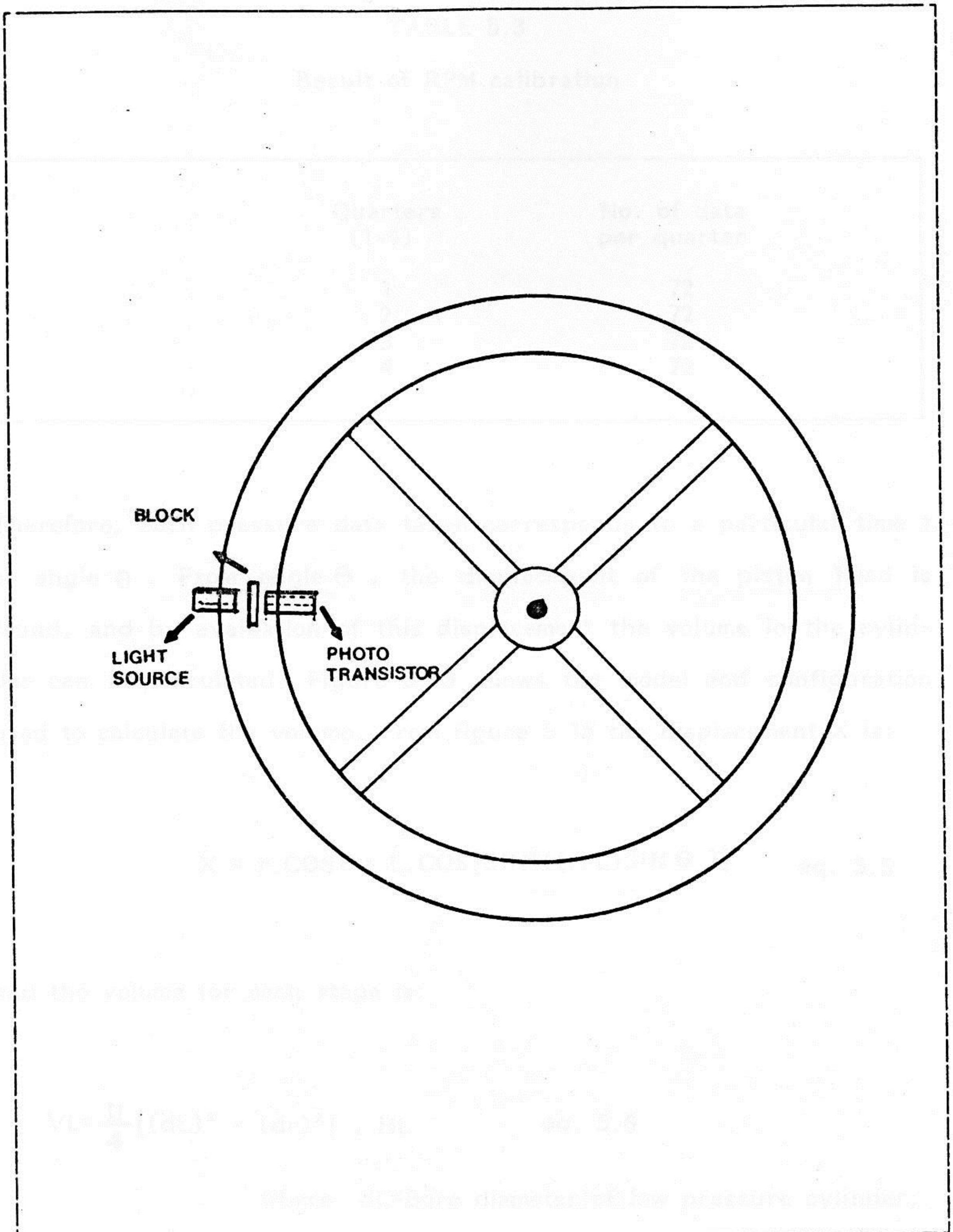


Figure 5.14: The flywheel and the position of photo transistor

TABLE 5.3

Result of RPM calibration

Quarters (1-4)	No. of data per quarter
1	72
2	72
3	72
4	72

Therefore, each pressure data taken corresponds to a particular time t or angle Θ . From angle Θ , the displacement of the piston head is found, and by evaluation of this displacement the volume in the cylinder can be calculated. Figure 5.15 shows the model and configuration used to calculate the volume. From figure 5.15 the displacement X is:

$$X = r.\cos\Theta + L.\cos[\sin^{-1}((r/L)\sin\Theta)] \qquad \text{eq. 5.5}$$

and the volume for each stage is:

$$VL=\frac{\Pi}{4}[(dL)^2 - (dr)^2] \,.\, HL \qquad \text{eq. 5.6}$$

where dL=Bore diameter of low pressure cylinder;

dr=Piston rod diameter;

HL=Displacement of low pressure piston;

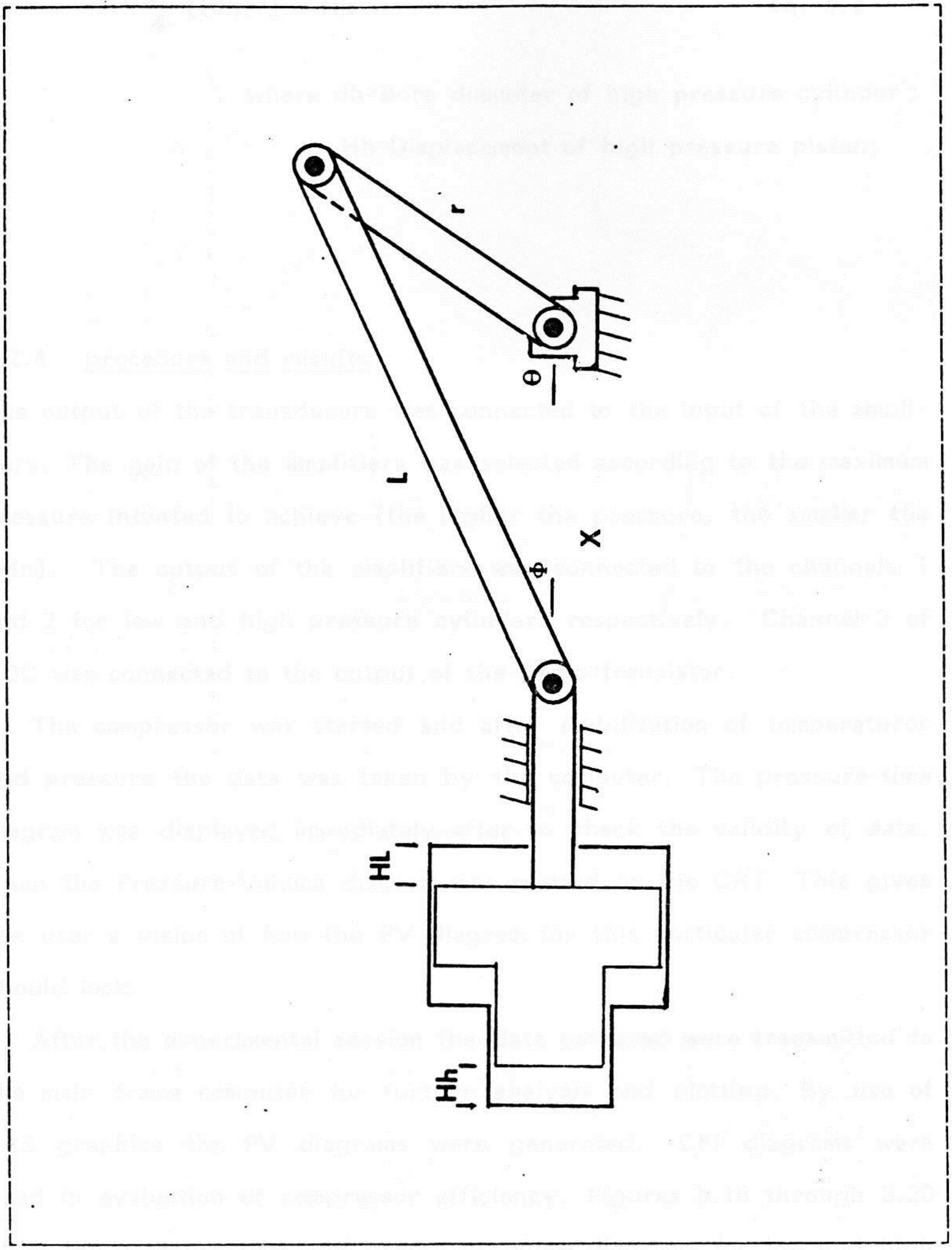


Figure 5.15: The configuration used for volume calculation

$$VH = \frac{\Pi}{4} [(dh)^2] \; . \; Hh \qquad \text{eq. 5.7}$$

where dh=Bore diameter of high pressure cylinder';

Hh=Displacement of high pressure piston;

5.2.4 procedure and results

The output of the transducers was connected to the input of the amplifiers. The gain of the amplifiers was selected according to the maximum pressure intented to achieve (the higher the pressure, the smaller the gain). The output of the amplifiers was connected to the channels 1 and 2 for low and high pressure cylinders respectively. Channel 3 of ADC was connected to the output of the photo transistor.

The compressor was started and after stabilization of temperatures and pressure the data was taken by the computer. The pressure-time diagram was displayed immediately after to check the validity of data. Then the Pressure-Volume diagram was plotted on the CRT. This gives the user a vision of how the PV diagram for this particular compressor should look.

After the experimental session the data gathered were transmitted to the main frame computer for further analysis and plotting. By use of SAS graphics the PV diagrams were generated. CPI diagrams were used in evaluation of compressor efficiency. Figures 5.16 through 5.20 show the pressure-time and pressure-volume diagrams for low and high pressure cylinders and pressure-volume for both stages on the same plot.

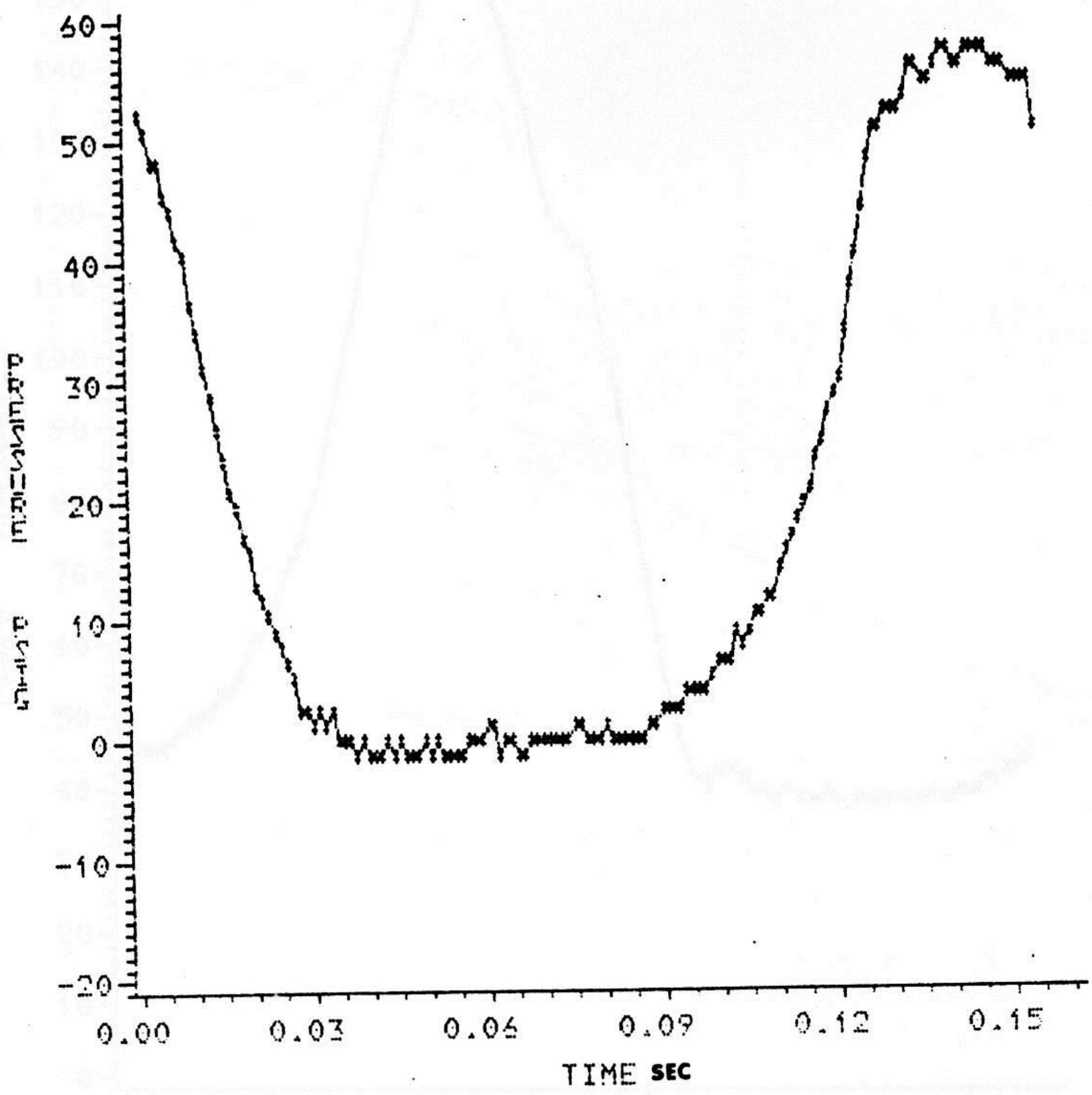


Figure 5.16: PT diagram for low pressure stage

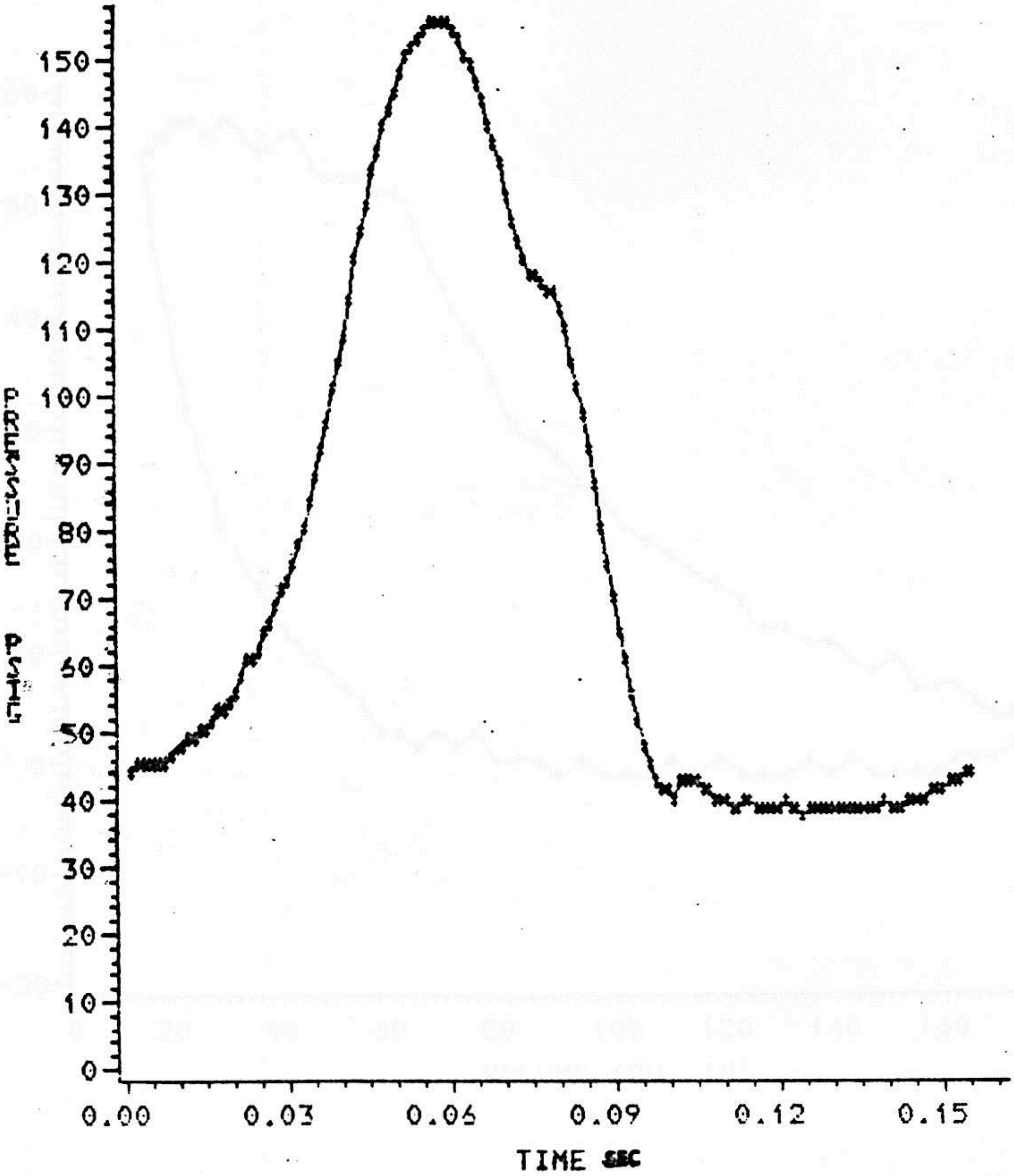


Figure 5.17: PT diagram for high pressure stage

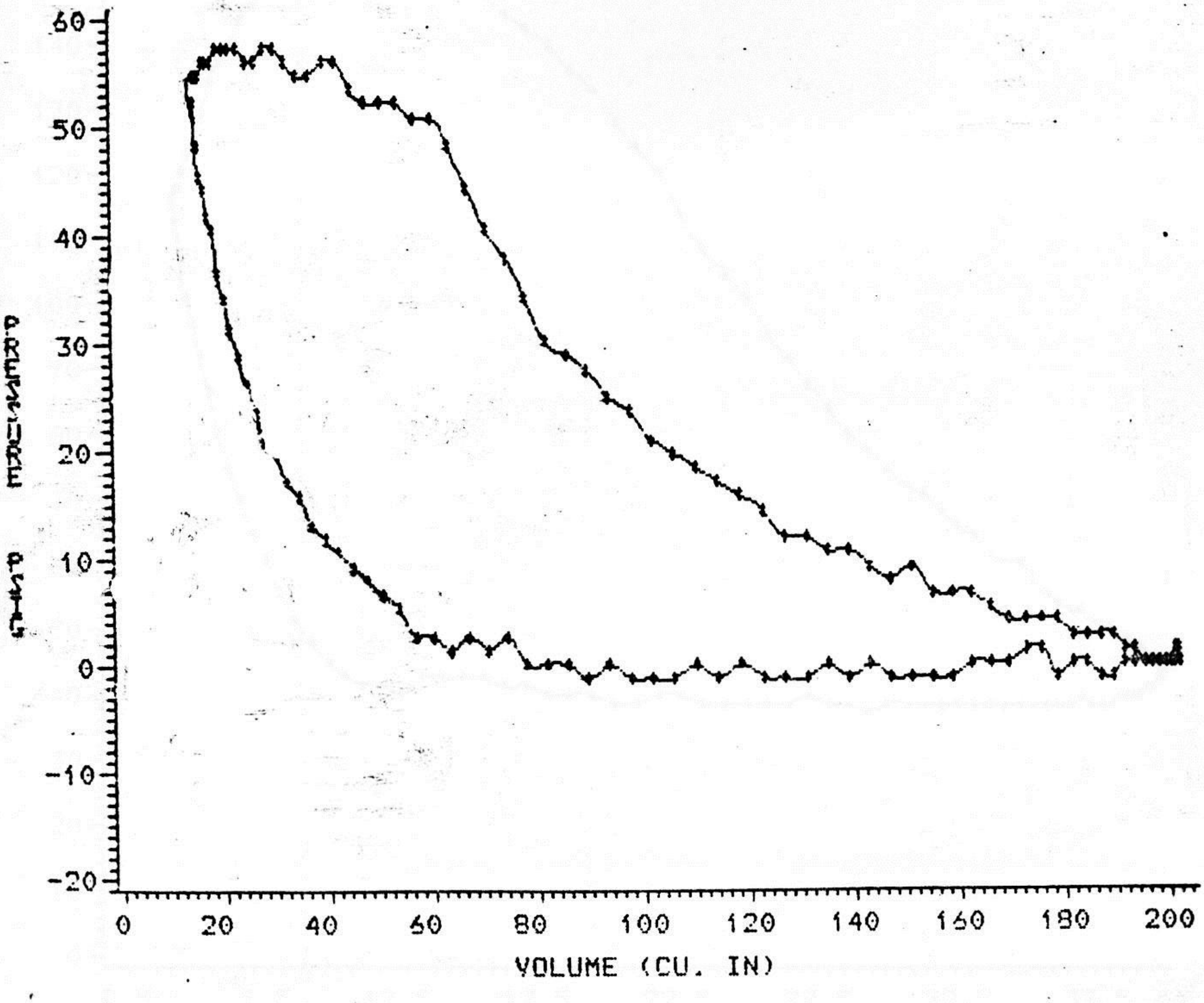


Figure 5.18: CPI diagram for low pressure stage.

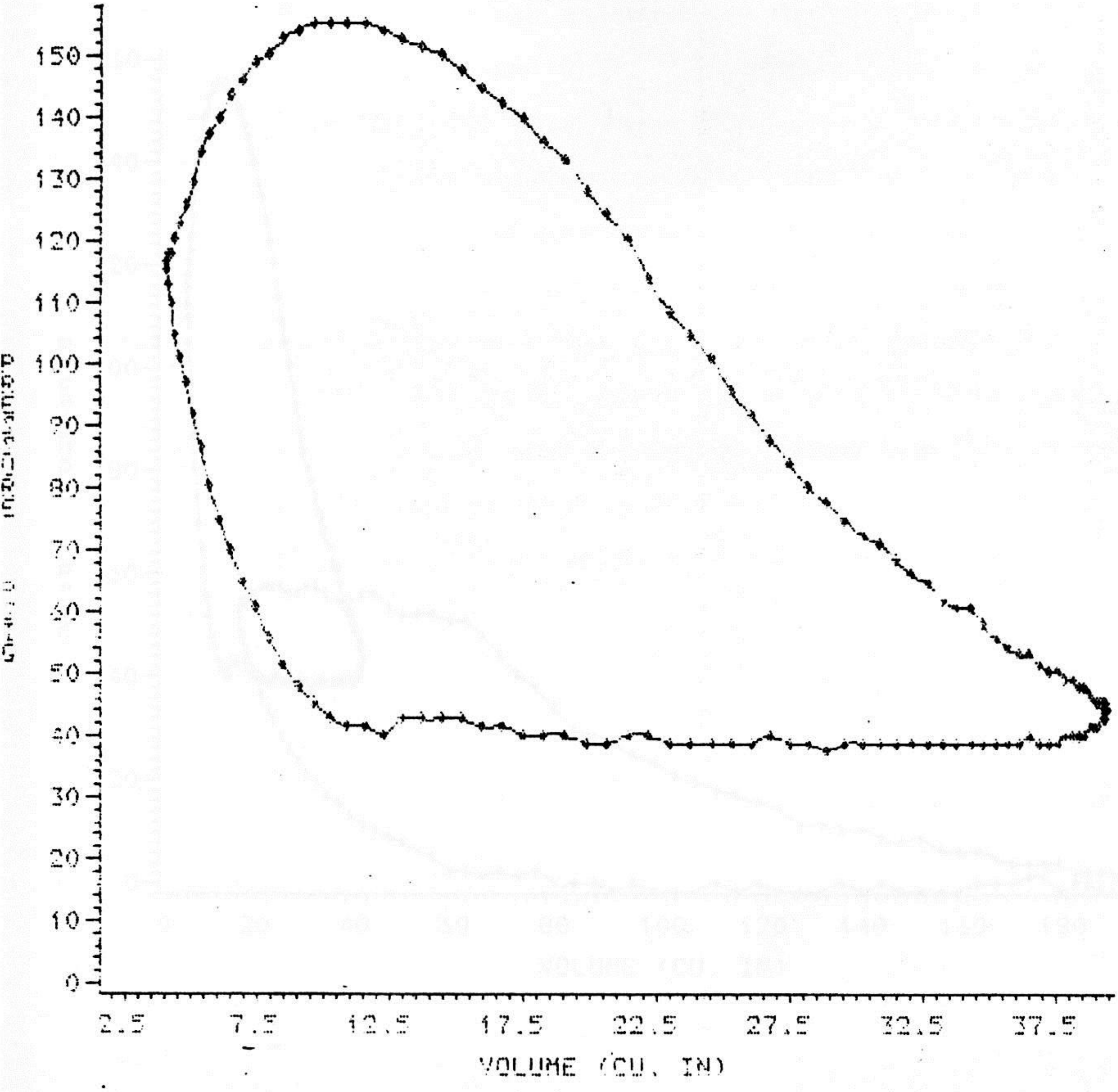


Figure 5.19: CPI diagram for high pressure stage

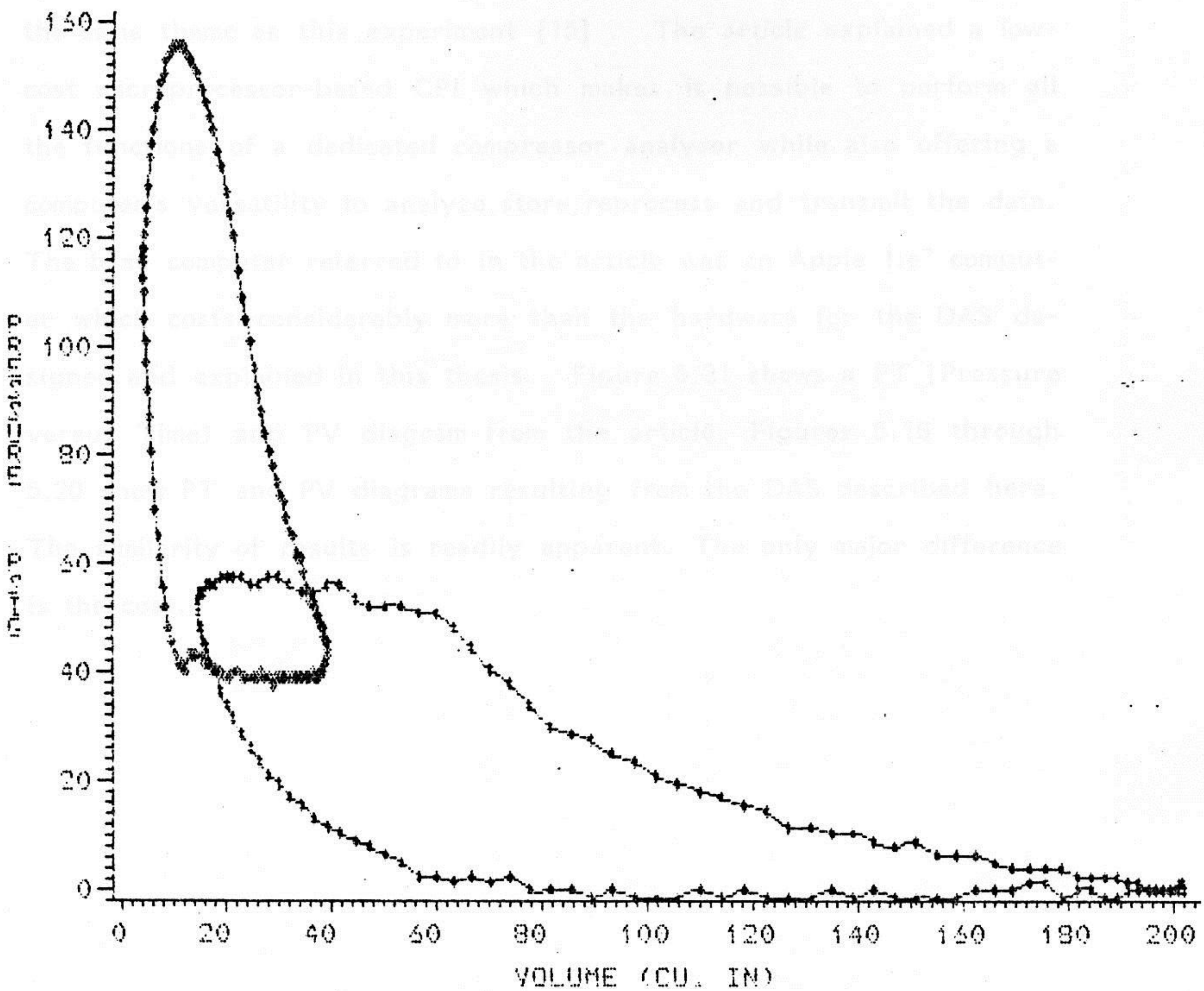


Figure 5.20: CPI diagram for the compressor (both stage)

The results from this experiment were satisfactory and proved to be valid. DAS also demonstrated its utility for educational experiments and even for some industrial research. For example in December, 1984 after this thesis project was completed, an article was published which had the same theme as this experiment [15] . The article explained a low-cost microprocessor-based CPI which makes it possible to perform all the functions of a dedicated compressor analyzer while also offering a computer's versatility to analyze, store, reprocess and transmit the data. The base computer referred to in the article was an Apple IIe[7] computer which costs considerably more than the hardware for the DAS designed and explained in this thesis. Figure 5.21 shows a PT (Pressure versus Time) and PV diagram from the article. Figures 5.16 through 5.20 show PT and PV diagrams resulting from the DAS described here. The similarity of results is readily apparent. The only major difference is the cost.

---

[7] Apple IIe is a registered trademark of Apple Computer, Inc.

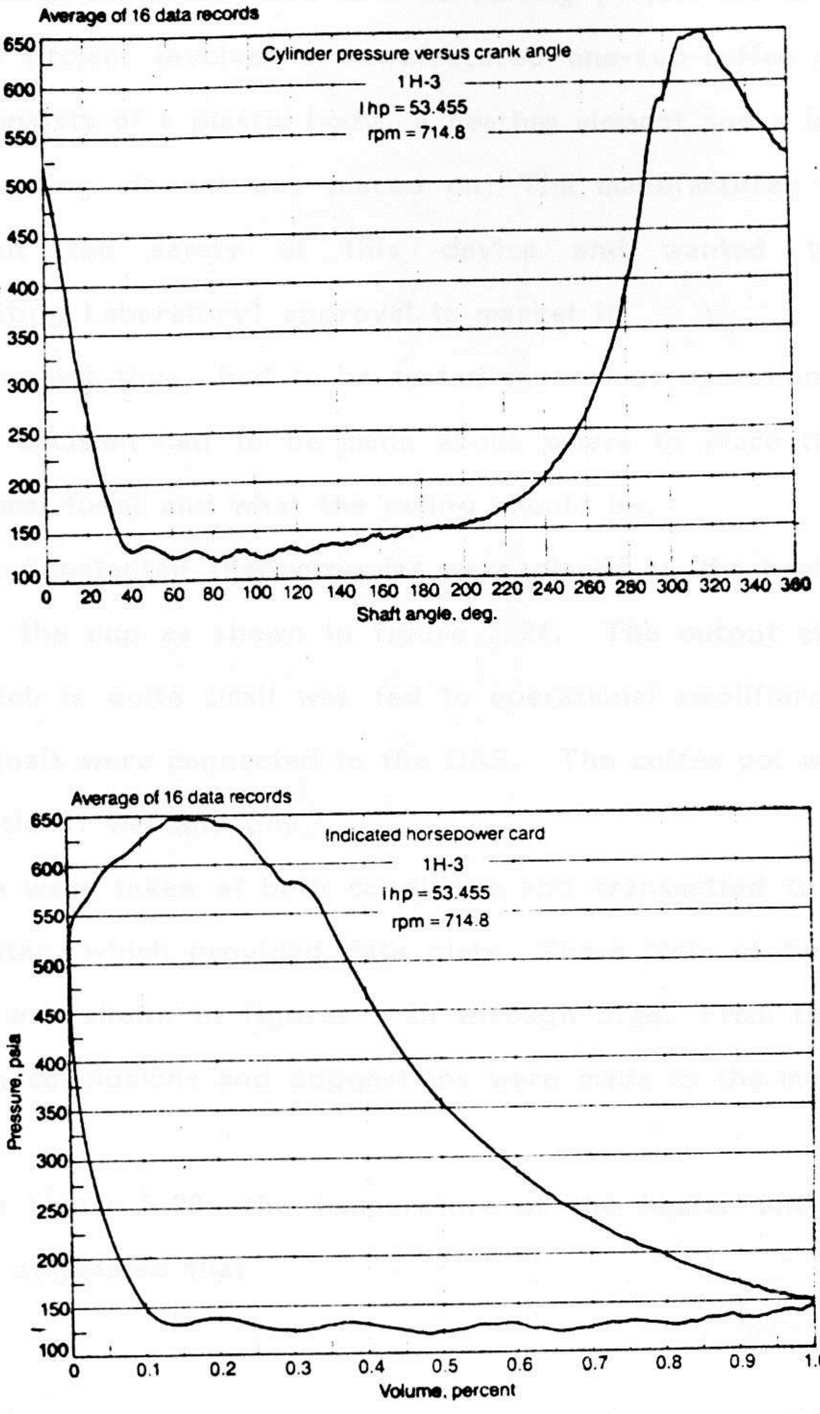


Figure 5.21: PT and PV diagram for a compressor (Adopted from Mechanical Engineering Magazine, DEC 1984 ,Page 67.)

5.3 Industrial application: Consulting project

The experiments explained here is a consulting project for a small industry. The project involves a manufactured one-cup-coffee pot. The coffee pot consists of a plastic body, a heating element and a base plate which the heating element was placed on. The manufacturer was concerned about the safety of this device and wanted to obtain UL(Underwriting Laboratory) approval to market it.

The coffee pot thus, had to be tested in various operational conditions and a decision had to be made about where to place the safety device (thermal fuse) and what the rating should be.

The Iron-Constantan thermocouples were placed on the heater,plate, and beneath the cup as shown in figure 5.24. The output of thermocouples, which is quite small was fed to operational amplifiers and the amplified signals were connected to the DAS. The coffee pot was tested in two conditions: wet and dry.

The data were taken at both conditions and transmitted to the main frame computer, which provided data plots. These plots of temperature versus time are shown in figures 5.25 through 5.28. From these plots the following conclusions and suggestions were made to the manufacturer.[8]

1. From figure 5.28, the temperature of the heater and plate, it was suggested that

---

[8] The author express his acknowlegements to the Dean of the Engineering and Architecture college, Dr. J. Stanislao, for initiating this project, and to Mr. B.W.Horton for his technical assistance.

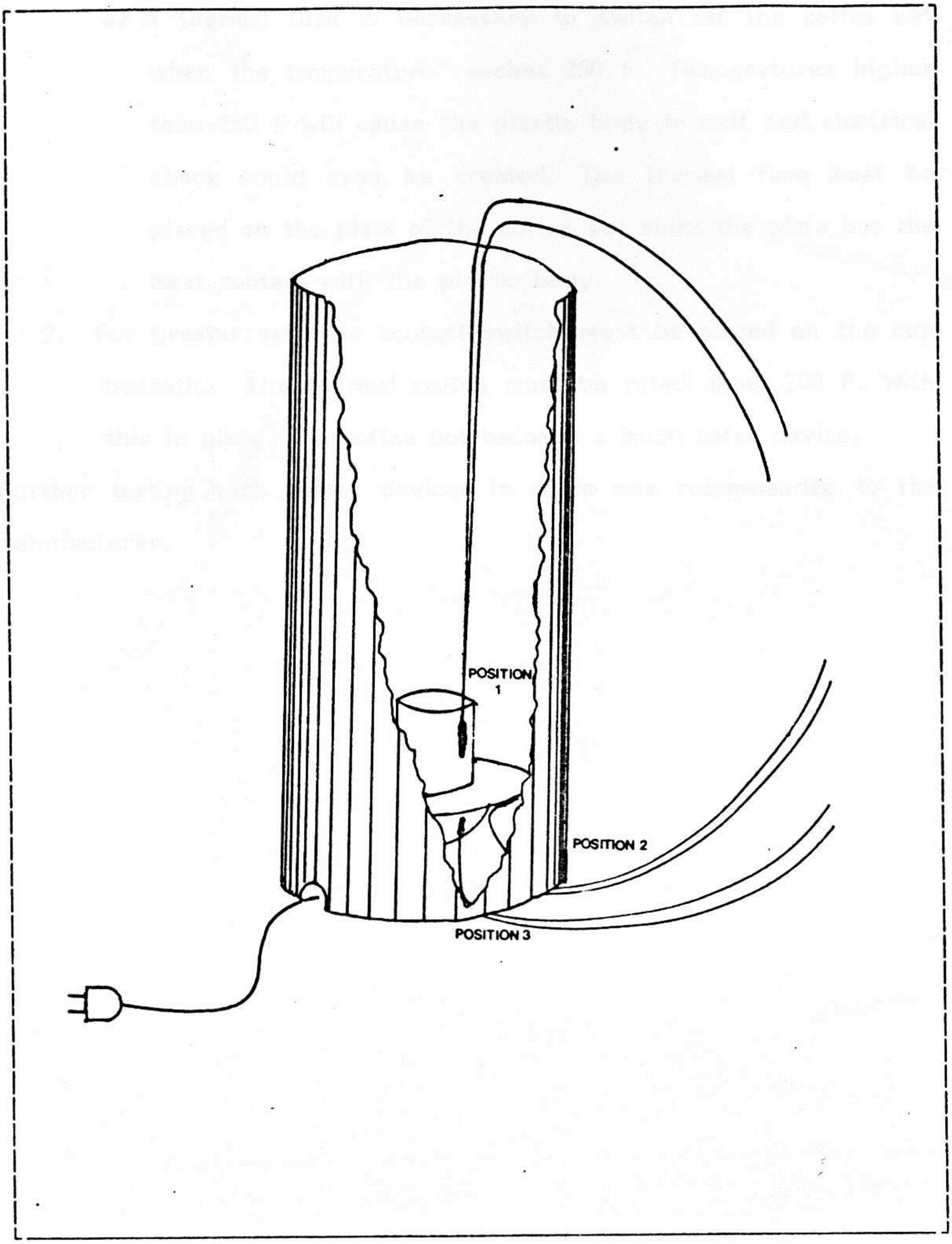


Figure 5.24: The coffee pot and the position of thermocouples

a) A thermal fuse is neccessary to switch off the coffee pot when the temperature reaches 250 F. Temperatures higher than 250 F will cause the plastic body to melt and electrical shock could even be created. The thermal fuse must be placed on the plate of the coffee pot since the plate has the most contact with the plastic body.

2. For greater safety a contact switch must be placed on the cup beneath. The thermal switch must be rated about 200 F. With this in place, the coffee pot becomes a much safer device.

Further testing with safety devices in place was recommended to the manufacturer.

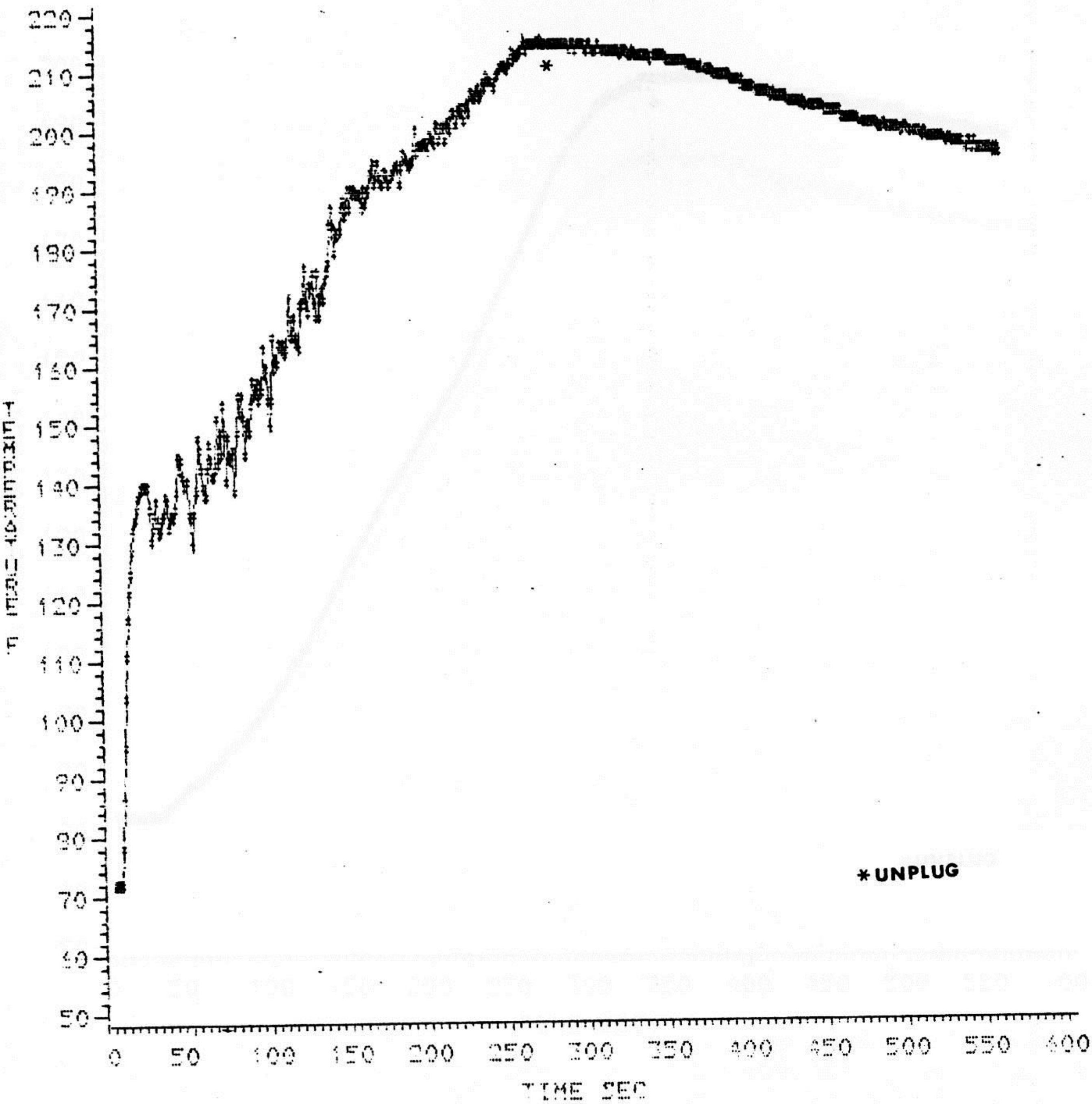


Figure 5.25: thermocouple 1 output against time (wet test)

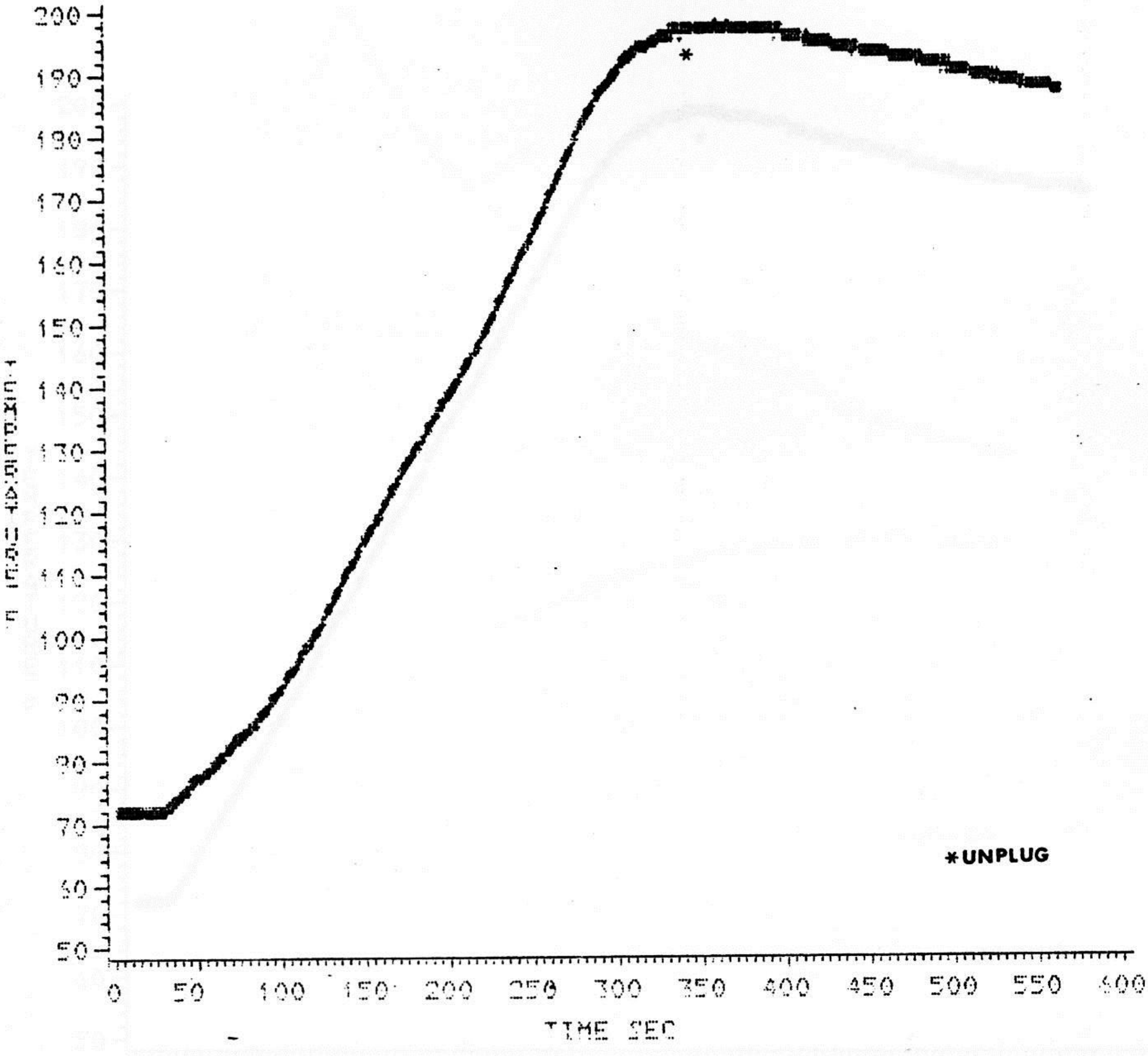


Figure 5.26: Thermocouple 2 output against time (wet test)

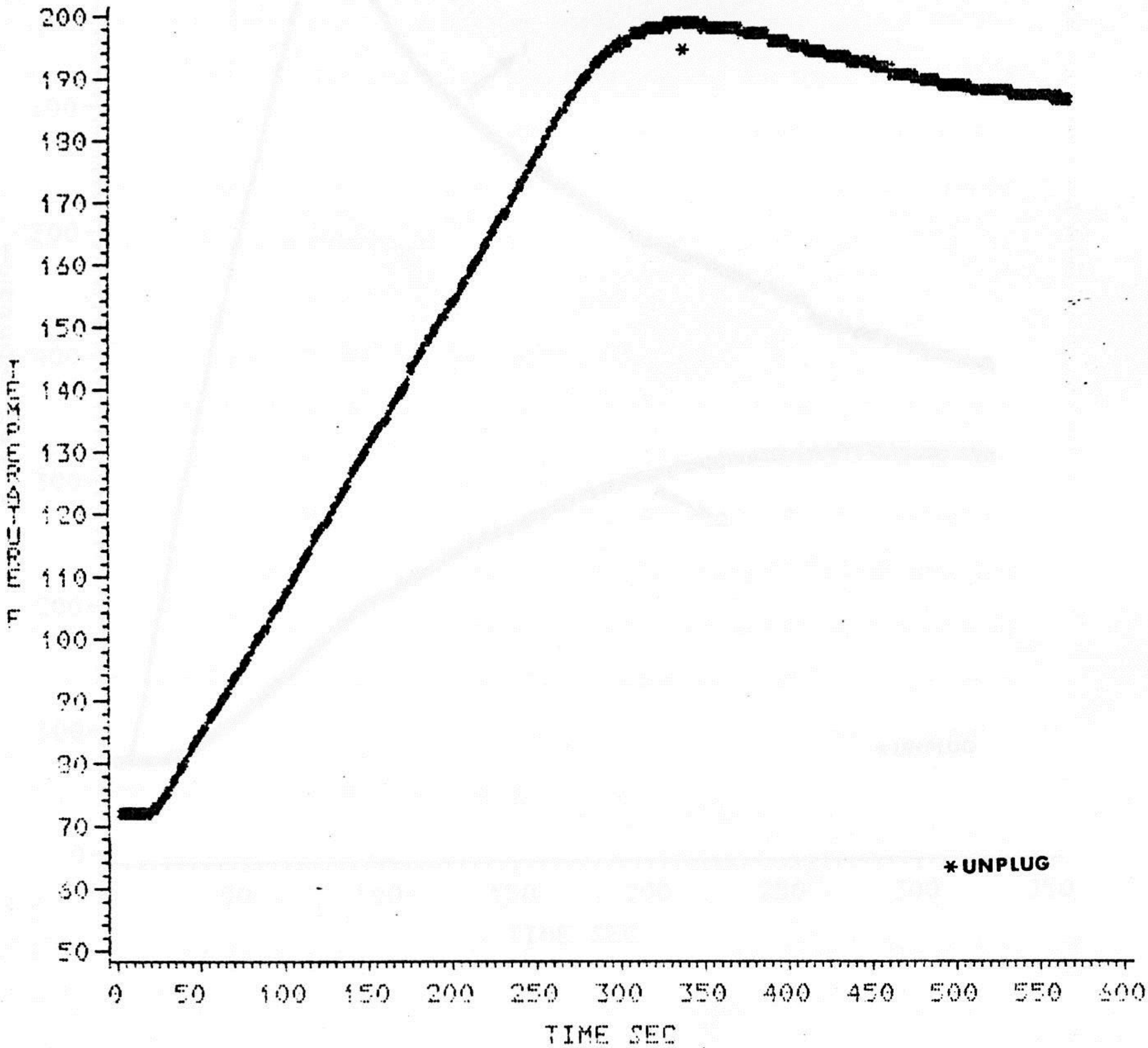


Figure 5.27: Thermocouple 3 output against time (wet test)

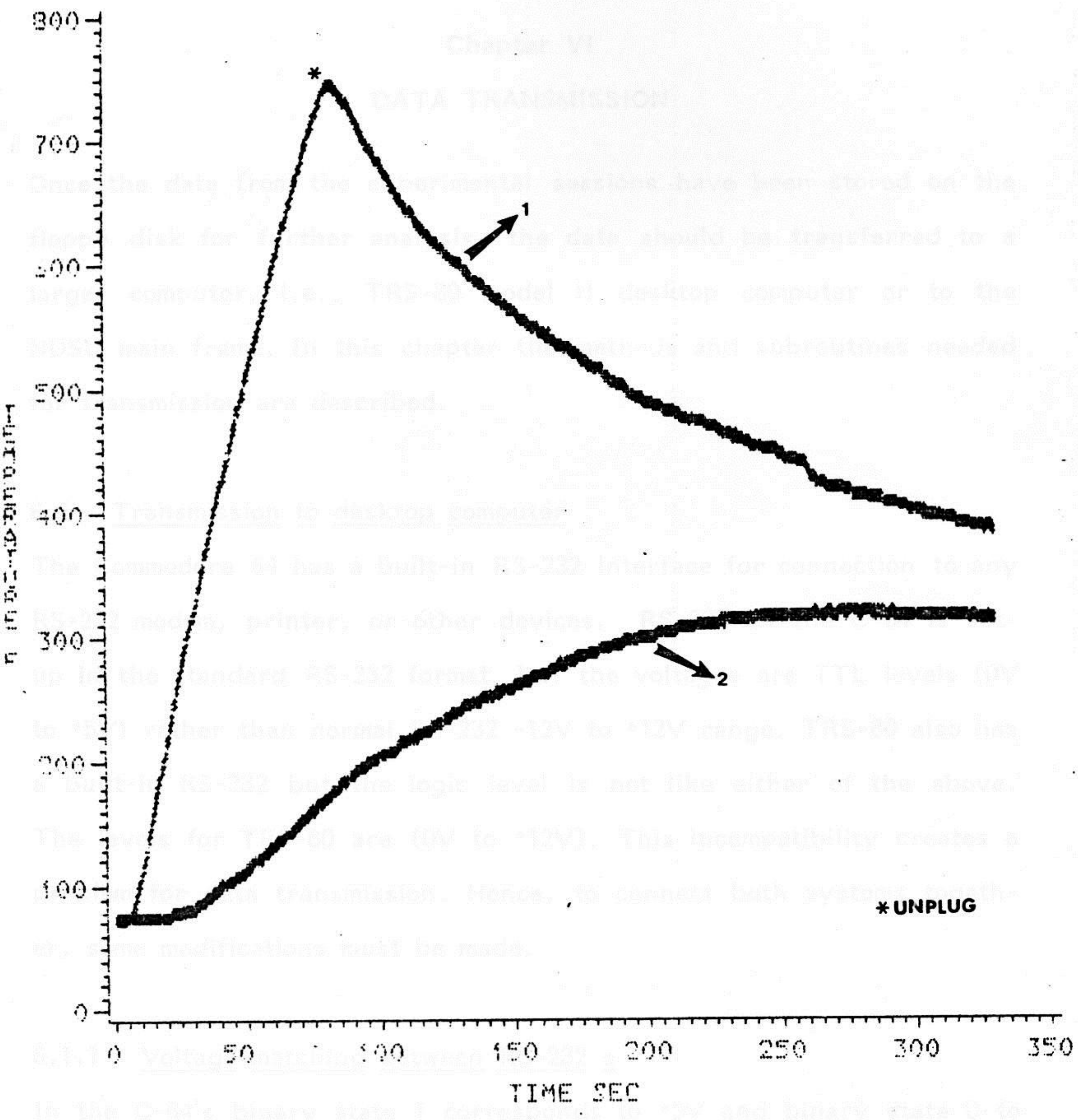


Figure 5.28: Thermocouple 1 and 2 outputs against time (dry test)

Chapter VI

DATA TRANSMISSION

Once the data from the experimental sessions have been stored on the floppy disk for further analysis, the data should be transferred to a larger computer, i.e., TRS-80 model II desktop computer or to the NDSU main frame. In this chapter the methods and subroutines needed for transmission are described.

6.1 Transmission to desktop computer

The Commodore 64 has a built-in RS-232 interface for connection to any RS-232 modem, printer, or other devices. RS-232 on the C-64 is set-up in the standard RS-232 format, but the voltages are TTL levels (0V to +5V) rather than normal RS-232 -12V to +12V range. TRS-80 also has a built-in RS-232 but the logic level is not like either of the above. The levels for TRS-80 are (0V to +12V). This incompatibility creates a problem for data transmission. Hence, to connect both systems together, some modifications must be made.

6.1.1 Voltage matching between RS-232 s

In the C-64's binary state 1 corresponds to +5V and binary state 0 to 0V at pin M of I/O port. On the other hand, at the RS-232 terminal of TRS-80 the binary state 1 corresponds to 0V and binary 0 to +12V. Therefore, these two computers are incompatiable in both voltage level

and polarity. This incompatibility can be resolved by a line driver MC1488 as shown on figure 6.1.

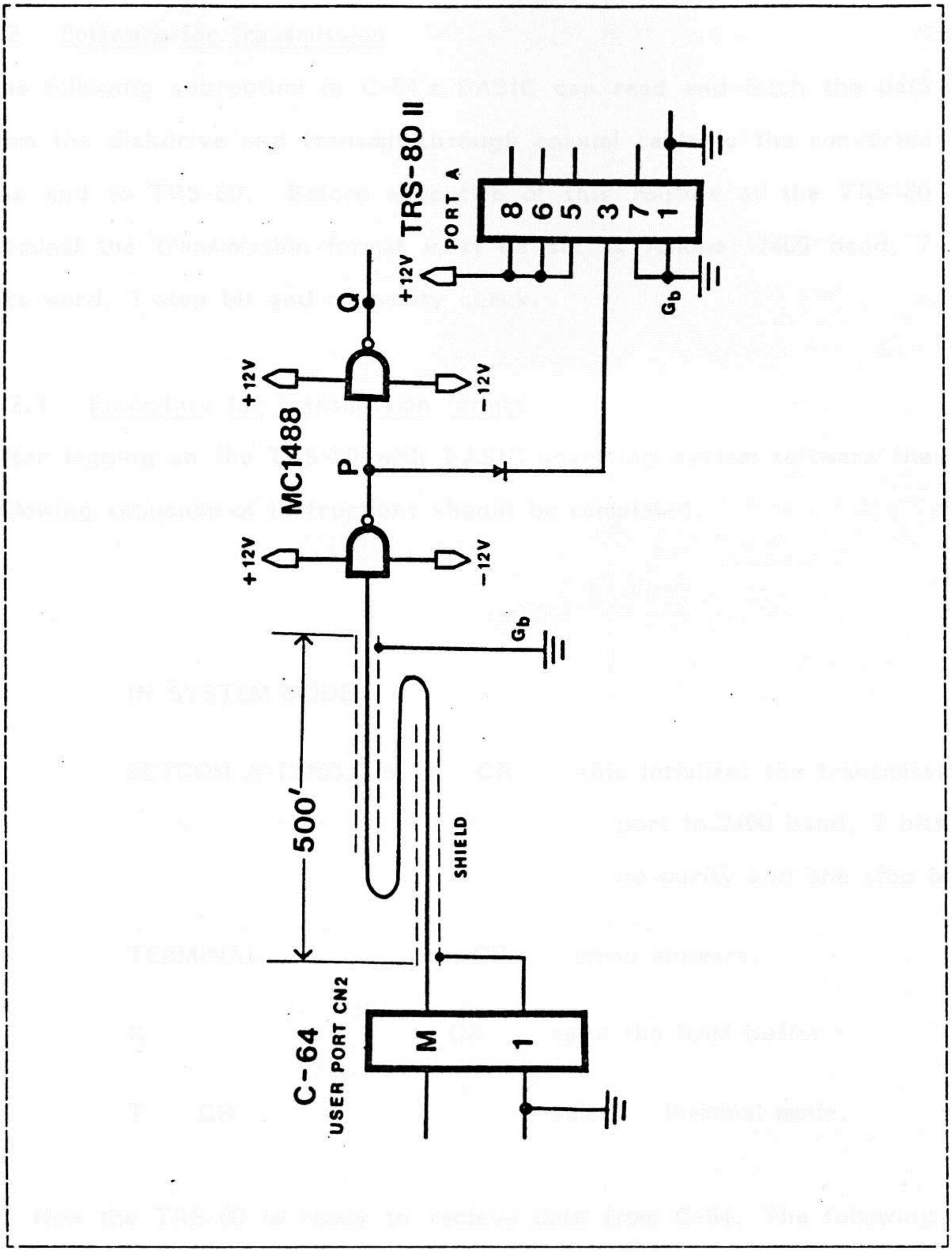


Figure 6.1: The line driver MC1488 implementation

6.2 Software for transmission

The following subroutine in C-64's BASIC can read and fetch the data from the diskdrive and transmit through coaxial cable to the converter box and to TRS-80. Before execution of this routine at the TRS-80 terminal the transmission format must be set as follows: 2400 baud, 7 bits word, 1 stop bit and no parity check.

6.2.1 Procedure for transmission format

After logging on the TRS-80 with BASIC operating system software the following sequence of instructions should be completed.

IN SYSTEM MODE:

SETCOM A=(2400,7,n,1)	CR	this intializes the transmission port to 2400 baud, 7 bits word no parity and one stop bit.
TERMINAL	CR	menu appears.
R	CR	open the RAM buffer.
T CR		select terminal mode.

Now the TRS-80 is ready to recieve data from C-64. The following is the transmission subroutine for C-64:

```
10 OPEN 2,2,3,CHR$(10+32)+CHR$(32+128): REM open a RS-232 channel
                                            with format of 2400 baud
                                            ,7 bits word, no parity
                                            and 1 stop bit.
20 FOR I=XXXXX TO YYYYY                 : REM XXXXX and YYYYY are the
                                            start and end address
                                            which the stored data
                                            are present.
30 PRINT#2,PEEK(I)                      : REM fetch the data at the
                                            address I and send to
                                            channel 2.
40 NEXT I
50 CLOSE 2                              : REM close the logical file.
```

As this program is completed, the data transmitted is stored in the TRS-80 RAM buffer as decimal numbers between 0 to 255. Then at the TRS-80 terminal the following instructions must be done.

BERAK Key		back to the menu.
R	CR	close the RAM buffer.
C	CR	copy RAM buffer onto disk.
FILENAME:		give a name to the data set.
....	CR	

Now the data is safely stored on disk for further analysis. If desiered, the data can be further transmitted from the TRS-80 to the main frame. To do so the data stored must be formatted. This is achieved by the following program.

Data sorting and formating program

N: Number of data lines to be transmitted

NAME1: Name of the data file containing unformatted data.

NAME2: Name of the data file which the formatted data to be saved.

```
10 OPEN "I",1,"NAME1"
20 OPEN "O",2,"NAME2"
30 FOR I=1 TO N                      repeat for number of lines;
40 FOR J=1 TO 10                     set 10 data per line;
50 INPUT#1,X                         remove data from file 1;
60 PRINT#2,X;                        put data to file 2 ;
70 NEXT J
80 PRINT#2,CHR$(13);                 put a carriage return
                                        ASCII character in file;
90 NEXT I
100 CLOSE                            close all files;
```

At this point the formatted data is stored on the disk. To transmit the data, the TRS-80 terminal must be reset.

IN SYSTEM MODE:		
SETCOM A=(300,7,E,1)	CR	set the communication parameters;
TERMINAL	CR	menu appears;
G	CR	get disk file into ram
FILE NAME:		buffer
....	CR	
W		set prompt wait character;
ENTER A NEW CHARACTER:		
- {SPACE BAR}		
Now go to terminal mode and log on the VSPC.		
Press BRAKE Key		menu appears;
X		transmit RAM buffer and enter terminal mode;

Now the data is transmitted and stored in the main frame computer.

6.3 Transmission to main frame computer

One of the accessories of the C-64 computer is the modem which can handle communications up to 300 baud rate. The NDSU main frame supports this communication speed. Software has been developed to directly transmit the acquired data to the main computer. The procedure for using this program is given in appendix J.

Chapter VII
EVALUATION OF THE SYSTEM AND CONCLUSION

The data acquisition in the past by and large relied on analog means. The data could be retained on the screen of a storage oscilloscope, a graph paper, a strip chart, and so on. When a quantitative analysis is required, the analog data had to be converted, by a human being, to digital data. Such conversion was not only inaccurate and inprecise but also quite tedious work. The invention of microprocessor and, subsequently, its availability at low cost has revolutionized the method of data acquisition. The modern microprocessor-based technology enables data acquisition in a digital form, thus eliminating subjective and often erroneous human intervention in the conversion process. The technology also makes data base analysis possible utilizing a microcomputer or a main frame computer. In addition, during the past few years, a new trend in instrumentation appeared above the horizon: the development of bus-oriented transducers, directly interfaceable to a computer system. There seems little doubt that computerization in data acquisition is an irreversible evolution. The ABET recommendation that the Mechanical Engineering Laboratories at NDSU be computerized reflect this evolution.

Commercially available data acquisition systems are expensive and at the present time beyond the departmental budget. It has been necessary for the Applied High-Tech Laboratory to develop a low cost proto-

type of data acquisition system which could be used in various laboratories in the Mechanical Engineering Department. This candidate was assigned by Dr.Okamura, thesis advisor, to develop hardware and software to convert Commodore 64, a low cost microcomputer,to a data acquisition system. The prototype should be quite versatile, menu driven, user-friendly and readily tailored to specifications of each laboratory. The system has been designed and successfully tested, and is already in use for various experiments in the Mechanical Engineering Laboratories. The system has many unique features which are not available in commercial units. An article about the system, coauthored by this candidate and Dr.Okamura, was published in the February, 1985, issue of BYTE, the "small systems journal" by McGraw-Hill. The authors have received many inquiries, request for software and consultation from industry, universities, research institutions and private persons across the United States, Canada, Mexico and Europe. This reflects the necessity of low cost data acquisition which is not commercially available.

As far as precision is concerned some may question whether eight bit resolution is high enough when the industrial standard is twelve bits. Many transducers and other instrumentation used in engineering laboratories are not accurate or precise enough to bother with twelve bits resolution. furthermore, the 8-bit ADC has a data conversion error of only 0.39% of full range, (i.e., 1/255) and this type of ADC is still being used in industries. The December,1984 issue of Mechanical Engineering [15] showed a similar system based on an APPLE computer to obtain CPI diagrams for the compressor which is basically what DAS

in this thesis has done.[9] A comparison of the output of DAS and the output of this system was given in chapter 5. Industrial-type data acquisition systems, e.g., TEXTRONICS and many other of commercial digital storage oscilloscope are based on 8-bit resolution ADC.

The system developed in this thesis is not recommended for a high presision research. However, it could serve well for experiments of an educational nature and in some cases for research as well, as long as all calibrations are done carefully and the range of errors are well understood.

7.1 Limitation of the system

As discussed above, one of the most often asked question is whether or not an 8-bit resolution of the ADC is enough, or if this system could be modified so that a 10 or 12-bit ADC could be implemented on the system? For most practical purposes, 8-bit resolution is high enough, but system can be used for 10 or 12-bit ADC. Of course, this would require new software and hardware arrangements.

Another issue is the speed of the system. Unfortunately the speed of the system is limited. This is due to speed of the ADC and C-64, which have a clock of 900 KHz and 1.02 MHz respectively, and software delay. It is possible for this system to become faster, but a new hardware is needed, namely, Direct Memory Access chip (DMA) and new software would have to be developed. Nevertheless, for most mechanical engineering experiments the sampling rate of 4360 samples per sec-

---

[9] The article in Mechanical Engineering magazine appeared after this thesis project had been completed and submitted for publication to BYTE.

ond is adequate.

7.2 Applications

Although the applications for the DAS are apparent for the compression and cam analysis, this system can be used for almost all the mechanical engineering laboratories such as:

1. Solar energy;
2. Wind power;
3. Thermistor (temperature control system);
4. Various stain-gage applications;
5. Tension and compression testing.

This system also can be used in an industrial environment. After publication of this system in <u>BYTE</u> [1], the authors have received many inquiries from governmental agencies, companies, educational institutions, research institutions, and hobbyists who intended to use DAS for other applications, including:

1. petro chemical research;
2. environmental monitoring systems;
3. aircraft companies;
4. chemical laboratories.
5. educational institutions research

7.3 Outlook for the DAS

The DAS system currently has not reached its full capacity. There are many feature which could be added to the system to expand its capability and these will be briefly outlined:

1. Using a 10 or 12-bit ADC would increase the accuracy and resolution of the system. The C-64 is capable of interfacing to 10 or 12-bit ADC's. Of course, the software and hardware need to be modified accordingly.
2. New features could be developed to assist system users and be compatible to industrial-type data acquisitions,such as,
   a) Allocation of more memory for data storage;
   b) Scorlling the plot of data on the high-resolution screen, so that more data points taken by DAS could be displayed on screen as a function of time; and
   c) a menu-driven transmission subroutine which could be capable of direct transmission to TRS-80 and/or communication with a main frame computer.
3. The data sampling rate could be increased to some extent by employing a DAM chip. This sampling rate is still limited to the speed of ADC chip. This also requires a sustantial modification of software and hardware.

Appendix A

CALIBRATION METHOD FOR VELOCITY TRANSDUCER

The experimental arrangement for calibration of the velocity transducer is shown in figure A.1. When the magnetic core is manually raised and released, it falls downward under the influence of gravity. Other factors affecting the motion may be the magnetic field created by the electric current in the coil by the motion of the core. Hence, the acceleration of the rod is represented by :

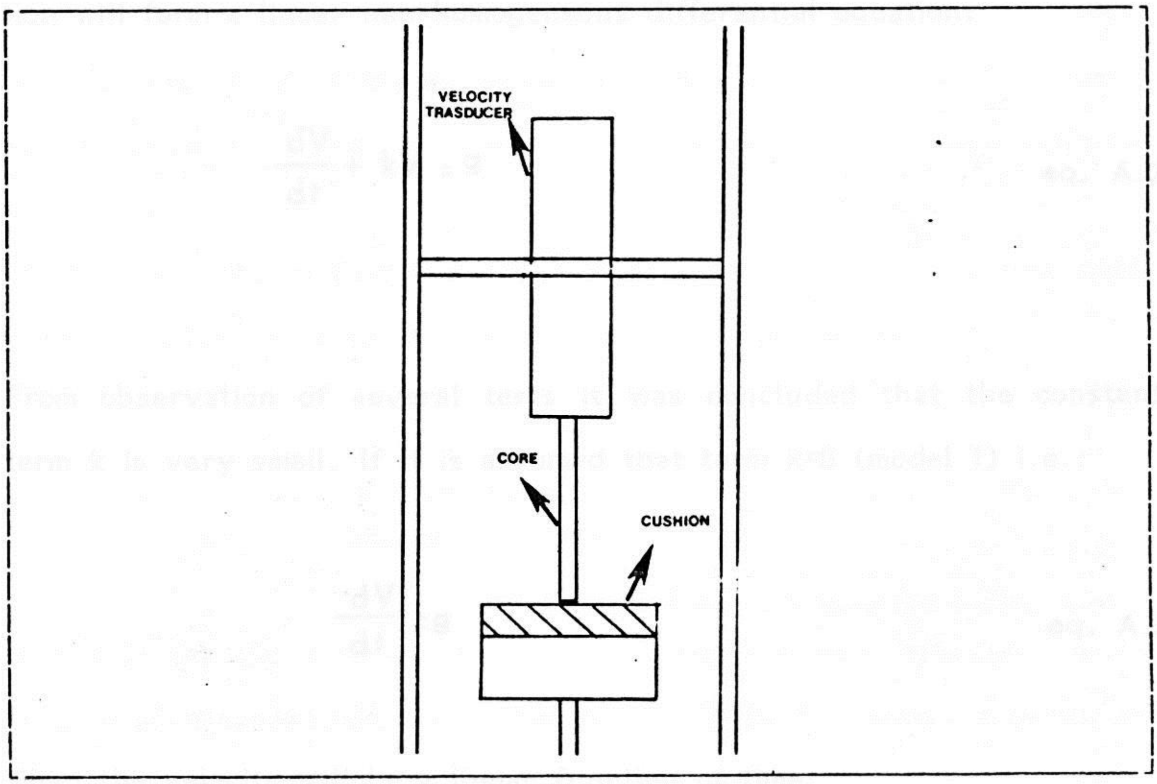


Figure A.1: Experimental setup

The acceleration of the rod from the figure is

$$a = g - kV \qquad \text{eq. A.1}$$

where V=velocity of core.

k=constant resistive factor.

g=gravitational acceleration.

a=acceleration of core.

Substituting acceleration of core by dV/dt and rearranging, the equation will form a linear non-homogeneous differential equation:

$$\frac{dV}{dt} + kV = g \qquad \text{eq. A.2}$$

From observation of several tests it was concluded that the constant term k is very small. If it is assumed that term k=0 (model 1) i.e.:

$$\frac{dV}{dt} = g \qquad \text{eq. A.3}$$

Then the velocity will be a linear function of time:

$$V = gt + C \qquad \text{eq. A.4}$$

Where C is an integral constant. The solution of Eq. A.2 with term k other than zero is (model 2):

$$V=\frac{g}{k}(1-e^{-kt}) \qquad \text{eq. A.5}$$

Equation A.5 is a non-linear function of time. While the statistical analysis for the model 1 (Eq. A.3) is straightforward, the statistical analysis for model 2 is more involved. By visual inspection, the data collected from the experiment appears to fit the linear model 1. Thus model 1 was applied to and fit the experimental data. The nonlinearity of these data were found from the linear model 1 and were considered as error. The intention of the calibration was to determine the sensitivity factor according to the experimental model 1. Examination of model 2 requires more extensive developement of theory, modeling and calibration method.

SAS was used to find the linear model which will describe the best linear model for the experimental data as presented below:

$$Y = -99.99\ X + b \qquad \text{eq. A.6}$$

Where X= Time;

Y= Output from the velocity transducer.

Term b in the above linear model is different for each data set due to the gain of the amplifier and initial bias. Figure A.2 shows a family of lines representing the data points and the linear lines after the model has been applied.

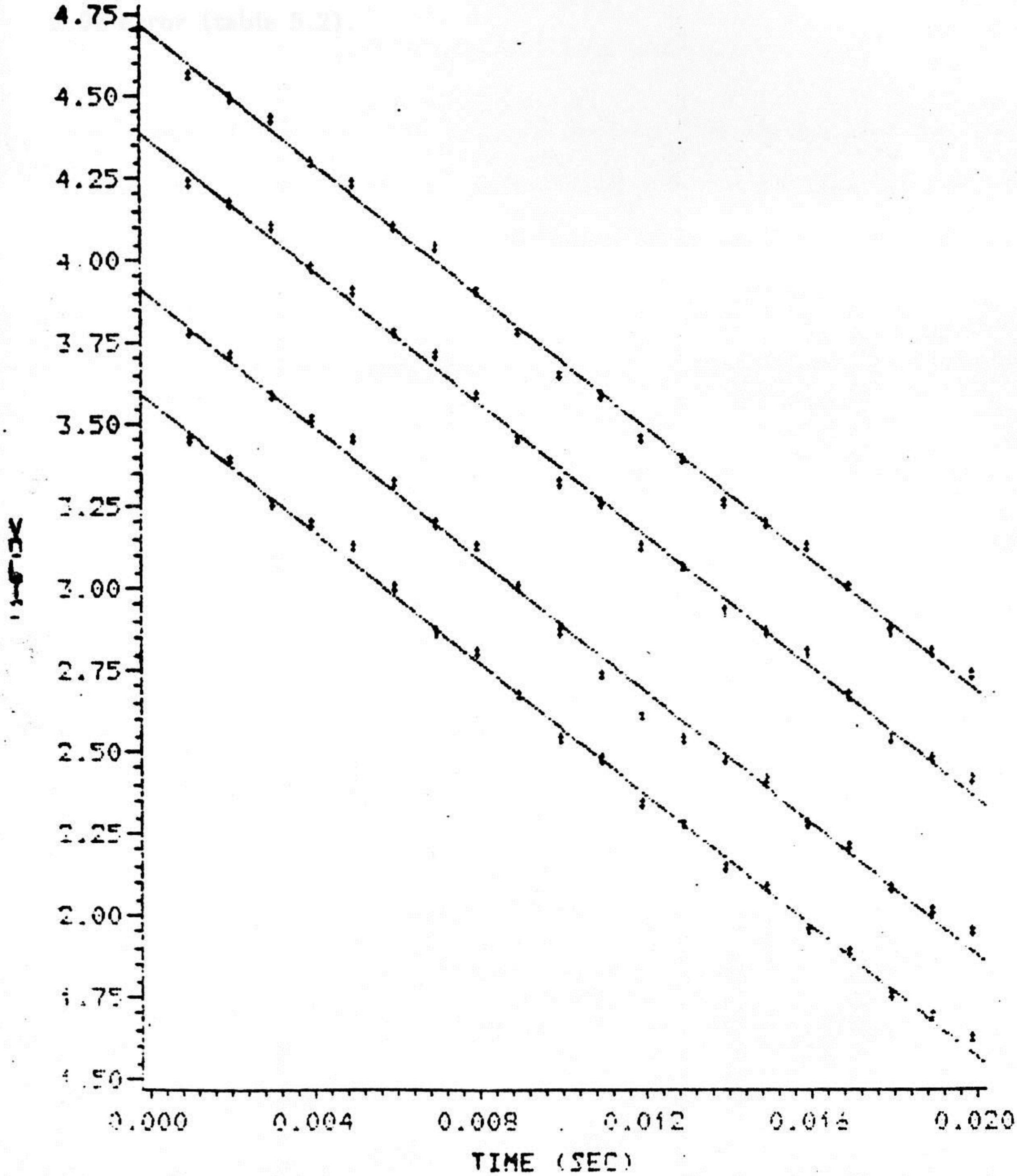


Figure A.2: Calibration curve

The nonlinearity was found to be about 2%. For educational purposes and many industrial applications, nonlinearity of up to 5% is tolerable. Applying the sensitivity factor to the actual velocity data showed only 2.9% error (table 5.2).

Appendix B

SOFTWARE FOR DAS (BASIC)

This appendix contains the listings of BASIC programs for two DAS systems, the general data acquisition and compressor data acquisition. The comments for each listing will follow after each program listings. (see [17],[18],[19],[20] for more information on software.)

General Data Acquisition Program

```
10 rem 340
15 poke55,255:poke56,125:clr
20 print""chr$(8)chr$(14)
25 poke53280,000:poke53281,000
30 poke53270,peek(53270)or16
40 printtab(5);"                                   "
50 printtab(5);"                                   "
60 printtab(5);"                                   "
70 printtab(5);"                                   "
80 printtab(5);"                                   "
90 printtab(5);"                                   "
100 printtab(5);"                                   "
110 printtab(11)" Data Acquisition "
120 printtab(11)" And Transmission "
140 printtab(18)"By:"
150 printtab(7)"Applied High-Tech Laboratory"
160 printtab(13)" Department of"
170 printtab(8)" Mechanical Engineering"
180 printtab(13)"Fargo,ND, 58105"
185 printtab(11);" version 1.0 (1984)"
190 printtab(13);"                 "
200 printtab(13);" Please Wait! "
210 printtab(13);"                 "
220 open 1,8,2,"0:listing2,s,r"
230 for i=50170 to 50271:input#1,a:pokei,a:next:close1
235 open 1,8,2,"0:listing3,s,r"
240 for i=49152 to 50161:input#1,a:pokei,a:next:close1
245 open 1,8,2,"0:listing4,s,r"
250 for i=50280 to 50398:input#1,a:pokei,a:next:close1
255 open 1,8,2,"0:listing5,s,r"
270 for i=50600 to 50618:input#1,a:pokei,a:next:close1
290 printtab(7);"                                 "
300 printtab(7)" PRESS ANY KEY TO CONTINUE "
310 printtab(7);"                                 "
320 poke2053,137:rem change the rem in the first line
330 getx$:ifx$=""then330
340 print""chr$(8)chr$(14)tab(17);" Menu "
345 poke53280,000:poke53281,000
350 printtab(17)" ____ "
360 printtab(9);" take Data in.........D"
370 printtab(9);" Plot on screen......P"
380 printtab(9);" Graph on printer....G"
390 printtab(9);" Transmit data.......T"
400 printtab(9);" Recall old data.....R"
410 printtab(9);" Store data..........S"
420 printtab(9);" Exit................E"
440 printtab(7);" ___________________________"
450 printtab(8);"                            "
460 printtab(8);" TYPE IN CHOICE REQUIRED "
470 printtab(8);"                            "
480 printtab(7);" ___________________________"
490 poke198,0:rem clear K/b buffer
```

```
500 getz$
510 if z$="d"goto590
520 if z$="p"goto790
530 if z$="g"goto1000
540 if z$="t"goto1110
550 if z$="s"goto1270
560 if z$="r"goto1470
570 if z$="e"goto1240
580 goto500
590 print""
600 poKe49240,22
610 input"->>>Number of channels(1-3)";aa:poKe767,aa
611 print"->>>Number of Data/channel is 320"
612 print"->>>Change? (Y/N)":inputr$:if r$="n" then 616
613 g=int(4096/aa):print"->>>Enter a new number(320-"g")":inputv
614 if v<320 or v>g then 613
615 goto 620
616 poKe198,0:v=320
620 K=int(v/256):poKe820,K:KK=v-K*256:poKe821,KK
621 print""tab(10);" Select sampling rate"
622 printtab(8);"1----->>>Default"
623 printtab(8);"2----->>>1000 sample/sec"
624 printtab(8);"3----->>>500  sample/sec"
625 printtab(8);"4----->>>100  sample/sec"
626 input x:if x<1 or x>4 then 626
627 on x goto 628,629,630,631
628 qq=001:ww=001:goto 639
629 qq=069:ww=001:goto 639
630 qq=055:ww=005:goto 639
631 qq=100:ww=017
639 a=aa-1:poKe50238,a:K=aa*v+32768:z=int(K/256):zz=K-(z*256)
640 poKe50212,zz:poKe50218,z
645 print""
650 printtab(8);"                          "
660 printtab(8);" COMPUTER IS IN PROCESS "
670 printtab(8);"                          "
680 poKe50260,qq
685 poKe50262,ww
690 poKe56334,peeK(56334)and254
700 poKe56579,0:poKe56323,255
710 poKe56321,2:sys(50170)
720 poKe56323,0
730 poKe56334,peeK(56334)or1
740 print""
749 printtab(6);"                          "
750 printtab(6);" PRESS ANY KEY TO CONTINUE "
755 printtab(6);"                          "
760 poKe198,0
770 getq$:ifq$=""then770
780 goto340
```

```
790 print"":aa=peek(767):ifaa>=1 and aa<=3then 800
792 print"->>ERROR....attempt to plot more than 3 or less than
794 getq$:if q$="" then794
796 goto 340
800 poke53270,peek(53270)and239:aa=peek(767):poke50601,4:poke49240,22
810 poke251,0:poke252,128
820 poke253,0:poke254,136
830 sys(50600)
840 poke53280,7
850 for i=679to700:pokei,0:next
860 b=aa*320+32768:t=int(b/256):tt=b-t*256
865 poke2,aa:poke759,aa
870 sys(49152)
880 sys(49229)
890 poke253,tt:poke254,t:poke2,aa:poke759,aa
900 sys(49274)
910 sys(49796)
920 getz$:ifz$=""then920
930 poke53265,peek(53265)and223
940 poke53272,21:poke56576,151:print""chr$(14)chr$(9)
950 poke53280,000:poke53281,000
955 poke53270,peek(53270)or16
960 poke251,0:poke252,136
970 poke253,0:poke254,128
980 sys(50600)
990 goto 340
1000 poke53265,peek(53265)and223
1005 print""tab(7)" DUMPING ON THE PRINTER"
1010 open4,4:cmd4:printchr$(8)
1020 sys(50280):printchr$(14)chr$(9):print#4 :close4
1030 poke50286,32
1040 poke53270,peek(53270)or16
1050 poke53280,0:poke53281,0
1070 printtab(15);"  OK LISTING"
1080 printtab(7);" PRESS ANY KEY TO CONTINUE "
1090 getc$:ifc$=""then1090
1100 goto340
1110 poke55,255:poke56,125:clr:print""
1120 printtab(8);"Switch to RS232 position"
1122 printtab(17);"AND"
1130 printtab(6);"  PRESS ANY KEY TO CONTINUE   "
1135 getz$:if z$="" then 1135
1136 print""
1140 open 2,2,3,chr$(10+32)+chr$(32+128):aa=peek(767)
1145 v=peek(820)*256+peek(821)
1150 fori=32769to32769+aa*v:d=peek(i)
1160 print#2,d;:print""d;:next:print#2,d
1170 close2:print"":printtab(8);"Switch to ADC position"
1180 printtab(17);"AND"
1190 printtab(5);" PRESS ANY KEY TO CONTINUE  "
1200 getz$:ifz$=""then1200
1210 poke55,255:poke56,125:clr
1220 poke53270,peek(53270)or16
1230 goto340
```

```
1240 poke53270,peek(53270)and239
1250 poke53272,21:poke56576,151:poke53280,254:poke53281,246:print"▓ ▓"
1260 printchr$(15)chr$(9):end
1270 print"▓"
1280 print"▓▓▓"
1300 printtab(6);"▓▓ Please insert the data disk!"
1320 printtab(5);"▓▓▓ please enter the name of the file":inputname$
1340 printtab(14);"▓▓▓▓▓▓▓Please Wait!"
1350 a$=chr$(34)+chr$(48)+chr$(58)
1360 b$=chr$(44)+chr$(83)+chr$(44)+chr$(87)+chr$(34)
1370 c$=a$+name$+b$:aa=peek(767)
1375 v=peek(820)*256+peek(821)
1380 open 4,8,4,c$:print#4,aa:print#4,v
1390 fori=1020to1023
1400 print#4,peek(i):next
1410 fori=32769to32769+aa*v
1420 print#4,peek(i):next
1430 close4
1440 printtab(7);"▓▓ PRESS ANY KEY TO CONTINUE "
1450 getx$:ifx$=""then1450
1460 goto340
1470 print"▓"
1480 print"▓▓▓▓▓"
1500 printchr$(18)tab(6);"▓ Please insert the data disk!"
1520 printtab(5);"▓ please enter the name of the file":inputname$
1530 printtab(14);"▓▓▓▓▓Please Wait!"
1540 a$=chr$(34)+chr$(48)+chr$(58)
1550 b$=chr$(44)+chr$(83)+chr$(44)+chr$(82)+chr$(34)
1560 c$=a$+name$+b$
1570 open 4,8,4,c$:input#4,aa:poke767,aa:input#4,v
1575 k=int(v/256):poke820,k:kk=v-k*256:poke821,kk
1580 fori=1020to1023
1590 input#4,a:pokei,a:next
1600 fori=32769to32769+aa*v
1610 input#4,a:pokei,a:next
1620 close4
1640 printtab(7);"▓▓ PRESS ANY KEY TO CONTINUE"
1660 getx$:ifx$=""then1660
1670 goto340
```

Comments for the General Data Acquisition program

Lines 10-210: First screen will be displayed as shown in figure B.1.

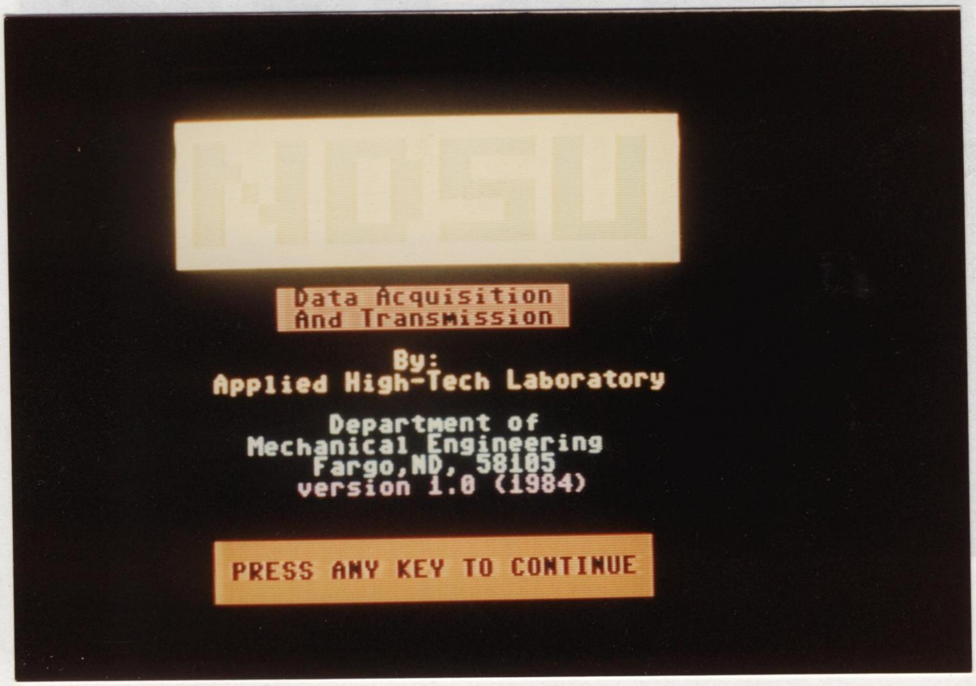


Figure B.1: First screen display

Lines 220-320: This is part of program will load the machine language routines form disk to memory of C64.

Lines 330-580: This part of program will display the main menu of the program as shown in figure B.2.

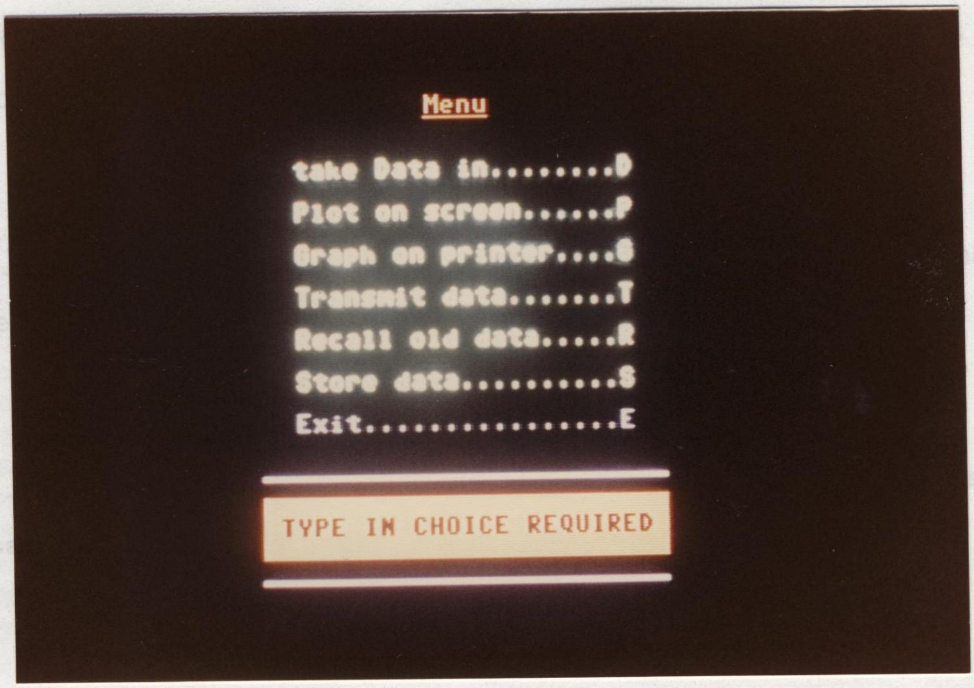


Figure B.2: Display of main menu

Lines 590-780: This part of program is where the data will be collected according to the number of channels and at the sampling rate chosen.

<u>Lines</u> <u>790-990</u>: This portion of program will input all the stored information to machine language and will plot the data collected on the high resolution screen.

<u>Lines</u> <u>1000-1100</u>: This will transfer the high-resolution screen to the dot matrix printer. (Note this must be done after the data has been plotted)

<u>Lines</u> <u>1110-1230</u>: The data collected and already stored in the memory will be serially transmitted to TRS-80 computer.

<u>Lines</u> <u>1240-1460</u>: The data stored in the memory of C64 will be stored as a sequential file on a disk.

<u>Lines</u> <u>1470-1670</u>: The already stored data will be read and stored in the memory of C64.

Compressor Data Acquisition program

```
10 rem 340
15 poke55,255:poke56,125:clr
20 print""chr$(8)chr$(14)
25 poke53280,000:poke53281,000
30 poke53270,peek(53270)or16
40 gosub 4000
340 print""chr$(8)chr$(14)tab(17);" Menu  "
345 poke53280,000:poke53281,000
350 printtab(17)" ‾‾‾‾"
360 printtab(9);" take Data in........D"
370 printtab(9);" Plot vs time........P"
380 printtab(9);" Graph on printer....G"
400 printtab(9);" Recall old data.....R"
406 printtab(9);" Bias................B"
407 printtab(9);" pV plot.............V"
408 printtab(9);" dIrectory...........I"
410 printtab(9);" Store data..........S"
420 printtab(9);" Exit................E"
450 printtab(8);"                              "
460 printtab(8);" TYPE IN CHOICE REQUIRED "
470 printtab(8);"                              "
490 poke198,0:rem clear K/b buffer
500 getz$
510 if z$="d"goto590
520 if z$="p"goto790
540 if z$="g"goto1000
546 if z$="b"goto2000
550 if z$="s"goto1270
555 if z$="i"then7000
560 if z$="r"goto1470
561 if z$="v"goto3000
570 if z$="e"goto1240
580 goto500
590 print""
616 poke198,0:v=320
631 qq=010:ww=001
632 aa=3:poke767,aa
639 a=aa-1:poke50238,a:k=aa*v+32768:z=int(k/256):zz=k-(z*256)
640 poke50212,zz:poke50218,z
645 print""
660 printtab(8);" COMPUTER IS IN PROCESS "
680 poke50260,qq
685 poke50262,ww:poke50275,88:poke50276,96
686 poke50518,76:poke50519,249:poke50520,195:poke50169,120
700 poke56579,0:poke56323,255
710 poke56321,2:sys(50500)
720 sys(50275):poke56323,0
740 print""
750 printtab(6);" PRESS ANY KEY TO CONTINUE "
760 poke198,0
770 getq$:ifq$=""then770
780 goto340
```

```
790 print"■":aa=peek(767)
800 poke53270,peek(53270)and239:aa=peek(767):poke50471,4:poke49240,22
810 poke251,0:poke252,128
820 poke253,0:poke254,136
830 sys(50470)
840 poke53280,7
850 for i=679to700:pokei,0:next
860 b=aa*320+32768:t=int(b/256):tt=b-t*256
865 poke2,aa:poke759,aa
870 sys(49152)
880 sys(49229)
890 poke253,tt:poke254,t:poke2,aa:poke759,aa
900 sys(49274)
910 sys(49796)
911 for i=8192+2562to8192+2560+320step8
912 pokei,255:next
913 fori=8192+5124 to 8192+5120+320step8
914 pokei,255:next
915 y=8192:pokey+328,240:pokey+329,96:pokey+330,96:pokey+331,96:pokey+332,96
916 pokey+333,96:pokey+334,127:pokey+335,127:pokey+339,126:pokey+340,126
917 for i=y+344 to y+344+7:readm:pokei,m:next:restore
918 pokey+2888,231:pokey+2889,102:pokey+2890,102:pokey+2891,126:pokey+2892,126
919 pokey+2893,102:pokey+2894,102:pokey+2895,231:pokey+2899,126:pokey+2900,126
920 for i=y+2904toy+2904+7:readm:pokei,m:next:restore
929 getz$:ifz$=""then929
930 poke53265,peek(53265)and223
940 poke53272,21:poke56576,151:print"■"chr$(14)chr$(9)
950 poke53280,000:poke53281,000
955 poke53270,peek(53270)or16
960 poke251,0:poke252,136
970 poke253,0:poke254,128
980 sys(50470)
990 goto 340
1000 poke53265,peek(53265)and223
1005 print"■"tab(7)"■ DUMPING ON THE PRINTER"
1010 open4,4:cmd4:printchr$(8)
1020 sys(50280):printchr$(14)chr$(9):print#4 :close4
1030 poke50286,32
1040 poke53270,peek(53270)or16
1050 poke53280,0:poke53281,0
1070 printtab(15);"■■■ OK LISTING"
1080 printtab(7);"■■■ PRESS ANY KEY TO CONTINUE "
1090 getc$:ifc$=""then1090
1100 goto340
```

```
1100 goto340
1240 poke53270,peek(53270)and239
1250 poke53272,21:poke56576,151:poke53280,254:poke53281,246:print" "
1260 printchr$(15)chr$(9):end
1270 print""
1280 print""
1300 printtab(12);" insert the data disk!"
1320 printtab(8);"  enter the name of the file":inputname$
1340 printtab(14);"Please Wait!"
1350 a$=chr$(34)+chr$(48)+chr$(58)
1360 b$=chr$(44)+chr$(83)+chr$(44)+chr$(87)+chr$(34)
1370 c$=a$+name$+b$:aa=peek(767)
1371 gosub 6000
1375 v=320
1380 open 4,8,4,c$:print#4,aa:print#4,v
1390 fori=1020to1023
1400 print#4,peek(i):next
1410 fori=32769to32769+aa*v
1420 print#4,peek(i):next
1430 close4
1440 printtab(7);" PRESS ANY KEY TO CONTINUE "
1450 getx$:ifx$=""then1450
1460 goto340
1470 print""
1480 print""
1500 printchr$(18)tab(12);" insert the data disk!"
1520 printtab(8);" enter the name of the file":inputname$
1530 printtab(14);"Please Wait!"
1540 a$=chr$(34)+chr$(48)+chr$(58)
1550 b$=chr$(44)+chr$(83)+chr$(44)+chr$(82)+chr$(34)
1560 c$=a$+name$+b$
1561 gosub 6000
1570 open 4,8,4,c$:input#4,aa:poke767,aa:input#4,v
1575 k=int(v/256):poke820,k:kk=v-k*256:poke821,kk
1580 fori=1020to1023
1590 input#4,a:pokei,a:next
1600 fori=32769to32769+aa*v
1610 input#4,a:pokei,a:next
1620 close4
1640 printtab(7);" PRESS ANY KEY TO CONTINUE"
1660 getx$:ifx$=""then1660
1670 goto340
```

```
2000 print"■":poke56579,0:poke56323,255
2010 printtab(15)"Bias Control"
2020 printtab(5)"■■turn bias knobs on amps until"
2030 printtab(5)"numbers are slightly greater    "
2040 printtab(5)"than one.■■"
2050 printtab(5);"Low pressure        High pressure"
2060 print"■■■■■■■■■■■■     press any key to continue"
2070 print"■■■■■■■■■■■"
2080 poke56323,255:poke56321,0
2090 c1=peek(56577):c1=peek(56577)
2100 poke56321,1
2110 c2=peek(56577):c2=peek(56577)
2120 print"■"tab(7);c1;tab(25);c2
2125 for i=1 to 200:next
2130 printtab(7);"■                          "
2140 print"■■":poke56323,0
2150 geta$:if a$="" then 2080
2160 goto 340
3000 print"■":poke53270,peek(53270)and239
3010 poke53280,7:j=0
3011 sys(49152)
3012 sys(49229)
3020 for i=32771 to 32771+3*320step3
3021 if peek(i)>100 then 3030
3022 next
3030 f=i:j=15
3031 for i=f+15*3 to 32771+3*320step3
3032 if peek(i)>100 then 3034
3033 j=j+1:next
3034 e=i
3050 theta=2*π/j
3060 for i=0 to j
3070 theta=2*π/j+theta:alpha=π-theta
3080 s=2.5*sin(alpha)/11
3090 gam=atn(s/sqr(-s*s+1)):phi=theta-gam
3100 l=sqr(6.25+121-55*cos(phi))
3110 vl=π/4*(7↑2-1.125↑2)*(l-8.5)
3120 vh=π/4*(3↑2)*(5-(l-8.5))
3130 x=24+int(vl):poke251,x
3140 y=20+peek((f+1)+3*i)-peek(1020):poke253,y
3150 sys(50718)
3160 x=14+int(vh):poke251,x
3170 y=20+peek((f+2)+i*3)-peek(1021):poke253,y
3180 sys(50718)
3185 next
3186 for i=9312 to 9312+7:readm:pokei,m:next
3187 for i=9328 to 9328+7:readm:pokei,m:next:restore
3188 poke9323,126:poke9324,126
3190 getz$:ifz$=""then 3190
3200 poke53265,peek(53265)and223
3210 poke53272,21:poke56576,151
3220 poke53270,peek(53270)or16
3230 goto 340
```

```
4000 poKe2053,137
4001 gosub 6000
4010 load"mlcomp",8,1
4020 return
5000 print""
5010 poKe56579,0:poKe56323,255:poKe50443,128
5020 poKe50518,76:poKe50519,224:poKe50520,196
5030 poKe56321,2:sys(50500)
5040 poKe56323,0
5050 a=peeK(252):b=peeK(253)
5060 t1=((a+(b*255))/3600)
5070 rpm=20/t1
5080 printtab(9);"rpm="rpm
5090 geta$:ifa$=""then5090
5100 goto 340
6000 open15,8,15
6010 print#15,"i"
6020 close15:return
7000 gosub 6000
7010 open2,8,15
7020 print"":print"DISK DIRECTORY"
7030 open1,8,0,"$0"
7040 get#1,a$,b$
7045 get#1,a$,b$:get#1,a$,b$
7050 c=0
7060 if a$<>""then c=asc(a$)
7070 if b$<>""then c=c+asc(b$)*256
7080 print""mid$(str$(c),2);tab(3);"";
7090 get#1,b$:ifst<>0 then 7200
7100 ifb$<>chr$(34)then7090
7110 get#1,b$:ifb$<>chr$(34)thenprintb$;:goto7110
7120 get#1,b$:ifb$=chr$(32)then7120
7130 printtab(18);:c$=""
7140 c$=c$+b$:get#1,b$:ifb$<>""then7140
7150 print""left$(c$,3)
7160 ifst=0then 7045
7200 print"  BLOCK FREE"
7210 close1:close2
7220 printtab(6);" PRESS ANY KEY TO CONTINUE"
7230 get g$:ifg$=""then7230
7231 goto 340
10020 data252,198,198,198,252,224,224,224
10030 data231,102,102,102,102,126,60,24
```

Comments for the Compressor Data Acquisition program

Lines 10-580: Displays the main menu this is shown in figure B.3.

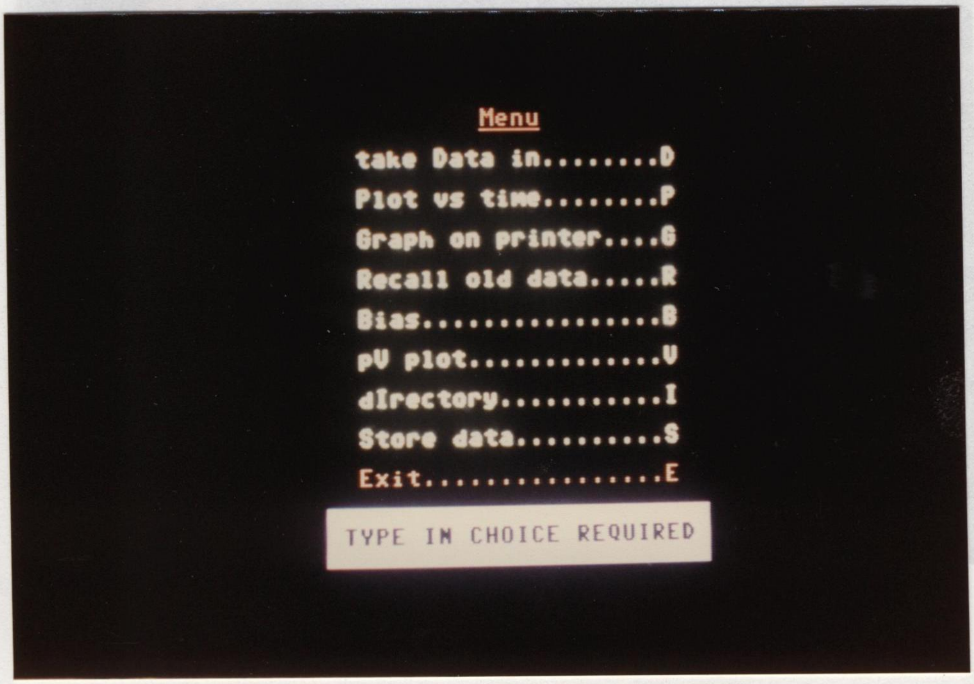


Figure B.3: Main menu for the compressor data acquisition

Lines 590-780: This portion of the program will collect data from the first three channels of the DAS.(first the low pressure, second the high pressure and third the photo transistor)

Lines 790-990: The collected data will be displayed on the high resolution screen. Figure B.4 show a sample of this plot.

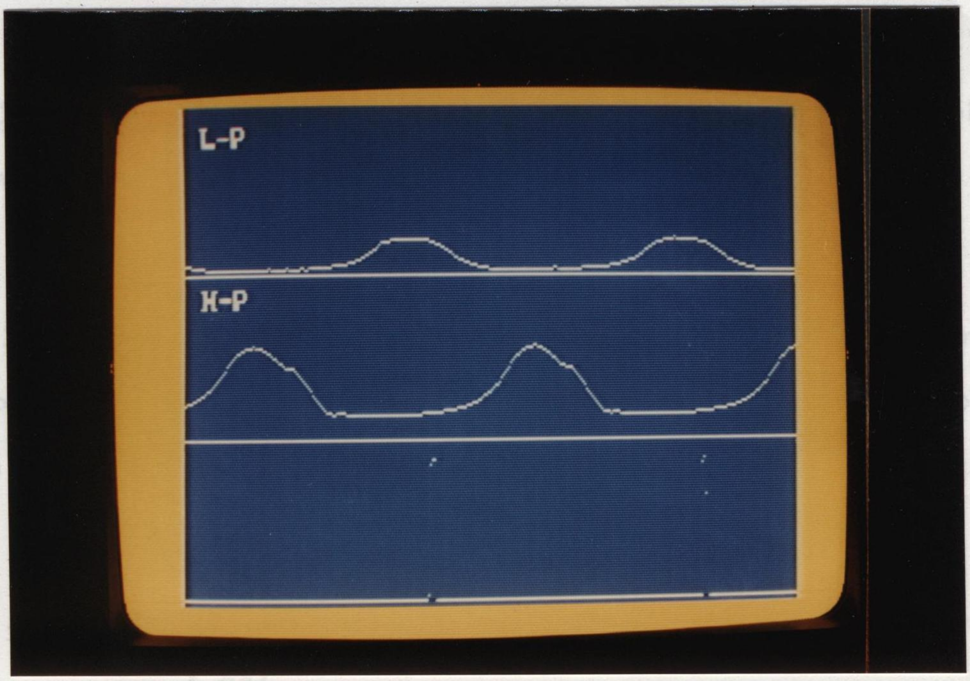


Figure B.4: A sample of pressure transducer and photo transistor output

Lines 1000-1100: This subroutine will transfer the high resolution screen to the dot matrix printer.

Lines 1240-1460: This subroutine will store the data stored in the memory on a disk.

Lines 1470-1670: This subroutine will read the data stored on a disk and stores them in the C64 memory.

Lines 2000-2160: This subroutine will monitor the bias of the amplifiers to allow the user to adjust the bias.

Lines 3000-3230: This portion of program will calculate the volume for each pressure data and will plot the PV diagram on the high resolution screen. Figure B.5 shows a sample of this plot.

Lines 4000-4020: A subroutine to load the machine language program into the memory of C64.

Lines 7010-10030: This subroutine will display the directory of the disk on the CRT .

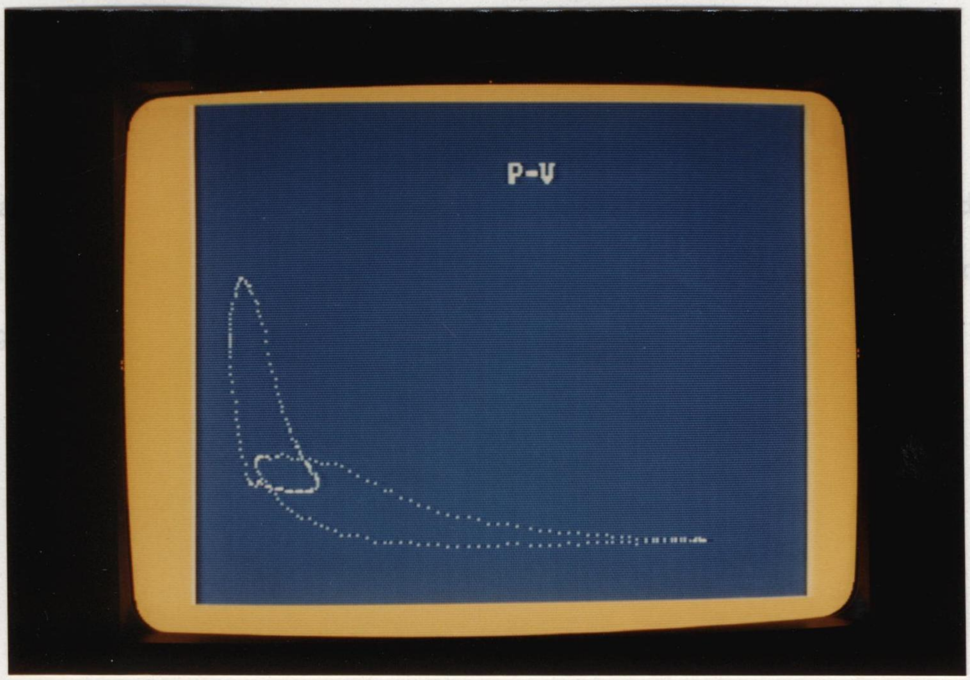


Figure B.5: Sample plot of the PV diagram on high resolution screen

Appendix C

ADC ML PROGRAM

This appendix contains the listing of assembly and instruction codes for the ADC routine. The comments for each block of codes will follow the listing.

The codes for Data collection

Address Decml	Hex	Machine Code	Assembly Code Program	
50170	C3FA	A9 00	LDAIM	0
50172	C3FC	85 FB	STAZ	251
50174	C3FE	A9 80	LDAIM	128
50176	C400	85 FC	STAZ	252
50178	C402	A9 00	LDAIM	0
50180	C404	85 FD	STAZ	253
50182	C406	A9 FF	LDAIM	255
50184	C408	8D 03 DC	STA	56323
50187	C40B	A9 00	LDAIM	0
50189	C40D	8D 03 DD	STA	56579
50192	C410	A5 FD	LDAZ	253
50194	C412	8D 01 DC	STA	56321
50197	C415	A2 19	LDXIM	25
50199	C417	CA	DEX	
50200	C418	D0 FD	BNE	253
50202	C41A	AD 01 DD	LDA	56577
50205	C41D	A2 00	LDXIM	0
50207	C41F	81 FB	STAIX	251
50209	C421	A5 FB	LDAZ	251
50211	C423	49 C0	EORIM	192
50213	C425	85 FE	STAZ	254
50215	C427	A5 FC	LDAZ	252
50217	C429	49 83	EORIM	131
50219	C42B	05 FE	ORAZ	254
50221	C42D	D0 01	BNE	1
50223	C42F	60	RTS	
50224	C430	18	CLC	
50225	C431	A9 01	LDAIM	1
50227	C433	65 FB	ADCZ	251
50229	C435	85 FB	STAZ	251
50231	C437	A9 00	LDAIM	0
50233	C439	65 FC	ADCZ	252
50235	C43B	85 FC	STAZ	252
50237	C43D	A9 02	LDAIM	2
50239	C43F	45 FD	EORZ	253
50241	C441	D0 07	BNE	7
50243	C443	A9 00	LDAIM	0
50245	C445	85 FD	STAZ	253
50247	C447	4C 4C C4	JMP	50252
50250	C44A	E6 FD	INCZ	253
50252	C44C	EA	NOP	
50253	C44D	EA	NOP	
50254	C44E	EA	NOP	
50255	C44F	EA	NOP	
50256	C450	EA	NOP	
50257	C451	EA	NOP	
50258	C452	EA	NOP	

```
50259 C453 | A2 01      | LDXIM      1
50261 C455 | A0 01      | LDYIM      1
50263 C457 | 88         | DEY
50264 C458 | D0 FD      | BNE      253
50266 C45A | CA         | DEX
50267 C45B | D0 F8      | BNE      248
50269 C45D | 4C 10 C4   | JMP    50192
```

Comments for data collection codes

Locations 50170-50191: Initialization of addresses and the Input/Output ports, i.e., CIA#1 port B set to output and CIA#2 port B all input.

Locations 50192-50201: Select the channel and some time delay for s

Locations 50202-50223: Read the data present at the port and store in the proper location. Check for last address if last address return from subroutine if not continue.

Locations 50224-50251: Adjust the address and the channel number.

Location 50252-50271: Time delay for sampling rate. Default value for locations 50260 and 50262 is 1. for different sampling rate values up to 255 can be selected.

Appendix D

HIGH-RESOLUTION PLOTTING ROUTINE

The following is the listing of the assembly and machine code written for High-Resolution plotting. This routine can plot up to three channels of data.

Listing for High-Resolution subroutine

Address Decml	Hex	Machine Code	Assembly Code Program	
49152	C000	A0 00	LDYIM	0
49154	C002	A2 00	LDXIM	0
49156	C004	A9 18	LDAIM	24
49158	C006	85 FB	STAZ	251
49160	C008	A9 D0	LDAIM	208
49162	C00A	85 FC	STAZ	252
49164	C00C	B1 FB	LDAIY	251
49166	C00E	09 08	ORAIM	8
49168	C010	91 FB	STAIY	251
49170	C012	A9 11	LDAIM	17
49172	C014	85 FB	STAZ	251
49174	C016	A9 D0	LDAIM	208
49176	C018	85 FC	STAZ	252
49178	C01A	B1 FB	LDAIY	251
49180	C01C	09 20	ORAIM	32
49182	C01E	91 FB	STAIY	251
49184	C020	A9 00	LDAIM	0
49186	C022	85 FB	STAZ	251
49188	C024	A9 20	LDAIM	32
49190	C026	85 FC	STAZ	252
49192	C028	A2 00	LDXIM	0
49194	C02A	A9 00	LDAIM	0
49196	C02C	81 FB	STAIX	251
49198	C02E	A9 3F	LDAIM	63
49200	C030	45 FB	EORZ	251
49202	C032	85 FD	STAZ	253
49204	C034	A9 3F	LDAIM	63
49206	C036	45 FC	EORZ	252
49208	C038	05 FD	ORAZ	253
49210	C03A	F0 10	BEQ	16
49212	C03C	18	CLC	
49213	C03D	A9 01	LDAIM	1
49215	C03F	75 FB	ADCZX	251
49217	C041	85 FB	STAZ	251
49219	C043	A9 00	LDAIM	0
49221	C045	75 FC	ADCZX	252
49223	C047	85 FC	STAZ	252
49225	C049	4C 28 C0	JMP	49192
49228	C04C	60	RTS	
49229	C04D	A9 00	LDAIM	0
49231	C04F	85 FB	STAZ	251
49233	C051	A9 04	LDAIM	4
49235	C053	85 FC	STAZ	252
49237	C055	A2 00	LDXIM	0
49239	C057	A9 16	LDAIM	22
49241	C059	81 FB	STAIX	251

Address Decml	Hex	Machine Code	Assembly Code Program	
49243	C05B	A9 E7	LDAIM	231
49245	C05D	45 FB	EORZ	251
49247	C05F	85 FD	STAZ	253
49249	C061	A9 07	LDAIM	7
49251	C063	45 FC	EORZ	252
49253	C065	05 FD	ORAZ	253
49255	C067	F0 10	BEQ	16
49257	C069	18	CLC	
49258	C06A	A9 01	LDAIM	1
49260	C06C	75 FB	ADCZX	251
49262	C06E	85 FB	STAZ	251
49264	C070	A9 00	LDAIM	0
49266	C072	75 FC	ADCZX	252
49268	C074	85 FC	STAZ	252
49270	C076	4C 55 C0	JMP	49237
49273	C079	60	RTS	
49274	C07A	A5 02	LDAZ	2
49276	C07C	49 01	EORIM	1
49278	C07E	F0 0C	BEQ	12
49280	C080	A5 02	LDAZ	2
49282	C082	49 02	EORIM	2
49284	C084	F0 09	BEQ	9
49286	C086	A5 02	LDAZ	2
49288	C088	49 03	EORIM	3
49290	C08A	F0 06	BEQ	6
49292	C08C	4C 95 C0	JMP	49301
49295	C08F	4C EE C0	JMP	49390
49298	C092	4C 93 C1	JMP	49555
49301	C095	A0 00	LDYIM	0
49303	C097	A9 01	LDAIM	1
49305	C099	85 F7	STAZ	247
49307	C09B	A9 80	LDAIM	128
49309	C09D	85 F8	STAZ	248
49311	C09F	B1 F7	LDAIY	247
49313	C0A1	8D A7 02	STA	679
49316	C0A4	A9 C8	LDAIM	200
49318	C0A6	8D A8 02	STA	680
49321	C0A9	A9 08	LDAIM	8
49323	C0AB	85 02	STAZ	2
49325	C0AD	A9 00	LDAIM	0
49327	C0AF	8D A9 02	STA	681
49330	C0B2	4E A8 02	LSR	680
49333	C0B5	90 04	BCC	4
49335	C0B7	18	CLC	
49336	C0B8	6D A7 02	ADC	679
49339	C0BB	6A	RORA	
49340	C0BC	6E A9 02	ROR	681
49343	C0BF	C6 02	DECZ	2
49345	C0C1	D0 EF	BNE	239
49347	C0C3	8D AA 02	STA	682

Address Decml	Hex	Machine Code	Assembly Code Program	
49350	C0C6	A9 08	LDAIM	8
49352	C0C8	85 02	STAZ	2
49354	C0CA	18	CLC	
49355	C0CB	6E AA 02	ROR	682
49358	C0CE	6E A9 02	ROR	681
49361	C0D1	C6 02	DECZ	2
49363	C0D3	D0 F5	BNE	245
49365	C0D5	18	CLC	
49366	C0D6	A9 C7	LDAIM	199
49368	C0D8	38	SEC	
49369	C0D9	ED A9 02	SBC	681
49372	C0DC	91 F7	STAIY	247
49374	C0DE	20 94 C3	JSR	50068
49377	C0E1	A0 00	LDYIM	0
49379	C0E3	AD F8 02	LDA	760
49382	C0E6	49 01	EORIM	1
49384	C0E8	F0 03	BEQ	3
49386	C0EA	4C 9F C0	JMP	49311
49389	C0ED	60	RTS	
49390	C0EE	A0 00	LDYIM	0
49392	C0F0	A9 01	LDAIM	1
49394	C0F2	85 F7	STAZ	247
49396	C0F4	A9 80	LDAIM	128
49398	C0F6	85 F8	STAZ	248
49400	C0F8	B1 F7	LDAIY	247
49402	C0FA	8D A7 02	STA	679
49405	C0FD	A9 64	LDAIM	100
49407	C0FF	8D A8 02	STA	680
49410	C102	A9 08	LDAIM	8
49412	C104	85 02	STAZ	2
49414	C106	A9 00	LDAIM	0
49416	C108	8D A9 02	STA	681
49419	C10B	4E A8 02	LSR	680
49422	C10E	90 04	BCC	4
49424	C110	18	CLC	
49425	C111	6D A7 02	ADC	679
49428	C114	6A	RORA	
49429	C115	6E A9 02	ROR	681
49432	C118	C6 02	DECZ	2
49434	C11A	D0 EF	BNE	239
49436	C11C	8D AA 02	STA	682
49439	C11F	A9 08	LDAIM	8
49441	C121	85 02	STAZ	2
49443	C123	18	CLC	
49444	C124	6E AA 02	ROR	682
49447	C127	6E A9 02	ROR	681
49450	C12A	C6 02	DECZ	2
49452	C12C	D0 F5	BNE	245
49454	C12E	18	CLC	
49455	C12F	A9 64	LDAIM	100

Address Decml	Hex	Machine Code	Assembly Code Program	
49457	C131	38	SEC	
49458	C132	ED A9 02	SBC	681
49461	C135	91 F7	STAIY	247
49463	C137	20 94 C3	JSR	50068
49466	C13A	A0 00	LDYIM	0
49468	C13C	AD F8 02	LDA	760
49471	C13F	49 01	EORIM	1
49473	C141	D0 01	BNE	1
49475	C143	60	RTS	
49476	C144	B1 F7	LDAIY	247
49478	C146	8D A7 02	STA	679
49481	C149	A9 64	LDAIM	100
49483	C14B	8D A8 02	STA	680
49486	C14E	A9 08	LDAIM	8
49488	C150	85 02	STAZ	2
49490	C152	A9 00	LDAIM	0
49492	C154	8D A9 02	STA	681
49495	C157	4E A8 02	LSR	680
49498	C15A	90 04	BCC	4
49500	C15C	18	CLC	
49501	C15D	6D A7 02	ADC	679
49504	C160	6A	RORA	
49505	C161	6E A9 02	ROR	681
49508	C164	C6 02	DECZ	2
49510	C166	D0 EF	BNE	239
49512	C168	8D AA 02	STA	682
49515	C16B	A9 08	LDAIM	8
49517	C16D	85 02	STAZ	2
49519	C16F	18	CLC	
49520	C170	6E AA 02	ROR	682
49523	C173	6E A9 02	ROR	681
49526	C176	C6 02	DECZ	2
49528	C178	D0 F5	BNE	245
49530	C17A	18	CLC	
49531	C17B	A9 C7	LDAIM	199
49533	C17D	38	SEC	
49534	C17E	ED A9 02	SBC	681
49537	C181	91 F7	STAIY	247
49539	C183	20 94 C3	JSR	50068
49542	C186	A0 00	LDYIM	0
49544	C188	AD F8 02	LDA	760
49547	C18B	49 01	EORIM	1
49549	C18D	F0 03	BEQ	3
49551	C18F	4C F8 C0	JMP	49400
49554	C192	60	RTS	
49555	C193	A0 00	LDYIM	0
49557	C195	A9 01	LDAIM	1
49559	C197	85 F7	STAZ	247
49561	C199	A9 80	LDAIM	128
49563	C19B	85 F8	STAZ	248

Address Decml Hex	Machine Code	Assembly Code Program
49565 C19D	B1 F7	LDAIY 247
49567 C19F	8D A7 02	STA 679
49570 C1A2	A9 42	LDAIM 66
49572 C1A4	8D A8 02	STA 680
49575 C1A7	A9 08	LDAIM 8
49577 C1A9	85 02	STAZ 2
49579 C1AB	A9 00	LDAIM 0
49581 C1AD	8D A9 02	STA 681
49584 C1B0	4E A8 02	LSR 680
49587 C1B3	90 04	BCC 4
49589 C1B5	18	CLC
49590 C1B6	6D A7 02	ADC 679
49593 C1B9	6A	RORA
49594 C1BA	6E A9 02	ROR 681
49597 C1BD	C6 02	DECZ 2
49599 C1BF	D0 EF	BNE 239
49601 C1C1	8D AA 02	STA 682
49604 C1C4	A9 08	LDAIM 8
49606 C1C6	85 02	STAZ 2
49608 C1C8	18	CLC
49609 C1C9	6E AA 02	ROR 682
49612 C1CC	6E A9 02	ROR 681
49615 C1CF	C6 02	DECZ 2
49617 C1D1	D0 F5	BNE 245
49619 C1D3	18	CLC
49620 C1D4	A9 42	LDAIM 66
49622 C1D6	38	SEC
49623 C1D7	ED A9 02	SBC 681
49626 C1DA	91 F7	STAIY 247
49628 C1DC	20 94 C3	JSR 50068
49631 C1DF	A0 00	LDYIM 0
49633 C1E1	AD F8 02	LDA 760
49636 C1E4	49 01	EORIM 1
49638 C1E6	D0 01	BNE 1
49640 C1E8	60	RTS
49641 C1E9	B1 F7	LDAIY 247
49643 C1EB	8D A7 02	STA 679
49646 C1EE	A9 42	LDAIM 66
49648 C1F0	8D A8 02	STA 680
49651 C1F3	A9 08	LDAIM 8
49653 C1F5	85 02	STAZ 2
49655 C1F7	A9 00	LDAIM 0
49657 C1F9	8D A9 02	STA 681
49660 C1FC	4E A8 02	LSR 680
49663 C1FF	90 04	BCC 4
49665 C201	18	CLC
49666 C202	6D A7 02	ADC 679
49669 C205	6A	RORA
49670 C206	6E A9 02	ROR 681
49673 C209	C6 02	DECZ 2

Address Decml	Hex	Machine Code	Assembly Code Program	
49675	C20B	D0 EF	BNE	239
49677	C20D	8D AA 02	STA	682
49680	C210	A9 08	LDAIM	8
49682	C212	85 02	STAZ	2
49684	C214	18	CLC	
49685	C215	6E AA 02	ROR	682
49688	C218	6E A9 02	ROR	681
49691	C21B	C6 02	DECZ	2
49693	C21D	D0 F5	BNE	245
49695	C21F	18	CLC	
49696	C220	A9 84	LDAIM	132
49698	C222	38	SEC	
49699	C223	ED A9 02	SBC	681
49702	C226	91 F7	STAIY	247
49704	C228	20 94 C3	JSR	50068
49707	C22B	A0 00	LDYIM	0
49709	C22D	AD F8 02	LDA	760
49712	C230	49 01	EORIM	1
49714	C232	D0 01	BNE	1
49716	C234	60	RTS	
49717	C235	B1 F7	LDAIY	247
49719	C237	8D A7 02	STA	679
49722	C23A	A9 42	LDAIM	66
49724	C23C	8D A8 02	STA	680
49727	C23F	A9 08	LDAIM	8
49729	C241	85 02	STAZ	2
49731	C243	A9 00	LDAIM	0
49733	C245	8D A9 02	STA	681
49736	C248	4E A8 02	LSR	680
49739	C24B	90 04	BCC	4
49741	C24D	18	CLC	
49742	C24E	6D A7 02	ADC	679
49745	C251	6A	RORA	
49746	C252	6E A9 02	ROR	681
49749	C255	C6 02	DECZ	2
49751	C257	D0 EF	BNE	239
49753	C259	8D AA 02	STA	682
49756	C25C	A9 08	LDAIM	8
49758	C25E	85 02	STAZ	2
49760	C260	18	CLC	
49761	C261	6E AA 02	ROR	682
49764	C264	6E A9 02	ROR	681
49767	C267	C6 02	DECZ	2
49769	C269	D0 F5	BNE	245
49771	C26B	18	CLC	
49772	C26C	A9 C6	LDAIM	198
49774	C26E	38	SEC	
49775	C26F	ED A9 02	SBC	681
49778	C272	91 F7	STAIY	247
49780	C274	20 94 C3	JSR	50068

Address Decml	Hex	Machine Code	Assembly Code Program	
49783	C277	A0 00	LDYIM	0
49785	C279	AD F8 02	LDA	760
49788	C27C	49 01	EORIM	1
49790	C27E	F0 03	BEQ	3
49792	C280	4C 9D C1	JMP	49565
49795	C283	60	RTS	
49796	C284	A9 01	LDAIM	1
49798	C286	85 F7	STAZ	247
49800	C288	A9 80	LDAIM	128
49802	C28A	85 F8	STAZ	248
49804	C28C	AD F7 02	LDA	759
49807	C28F	8D F6 02	STA	758
49810	C292	A0 00	LDYIM	0
49812	C294	A9 00	LDAIM	0
49814	C296	8D A8 02	STA	680
49817	C299	8D A9 02	STA	681
49820	C29C	AD A8 02	LDA	680
49823	C29F	29 F8	ANDIM	248
49825	C2A1	8D A7 02	STA	679
49828	C2A4	B1 F7	LDAIY	247
49830	C2A6	29 F8	ANDIM	248
49832	C2A8	8D AA 02	STA	682
49835	C2AB	A9 03	LDAIM	3
49837	C2AD	85 02	STAZ	2
49839	C2AF	A9 00	LDAIM	0
49841	C2B1	8D AB 02	STA	683
49844	C2B4	18	CLC	
49845	C2B5	0E AA 02	ASL	682
49848	C2B8	2E AB 02	ROL	683
49851	C2BB	C6 02	DECZ	2
49853	C2BD	D0 F5	BNE	245
49855	C2BF	AD AA 02	LDA	682
49858	C2C2	8D AC 02	STA	684
49861	C2C5	AD AB 02	LDA	683
49864	C2C8	8D AD 02	STA	685
49867	C2CB	A9 02	LDAIM	2
49869	C2CD	85 02	STAZ	2
49871	C2CF	18	CLC	
49872	C2D0	0E AC 02	ASL	684
49875	C2D3	2E AD 02	ROL	685
49878	C2D6	C6 02	DECZ	2
49880	C2D8	D0 F5	BNE	245
49882	C2DA	18	CLC	
49883	C2DB	AD AA 02	LDA	682
49886	C2DE	6D AC 02	ADC	684
49889	C2E1	8D AA 02	STA	682
49892	C2E4	AD AB 02	LDA	683
49895	C2E7	6D AD 02	ADC	685
49898	C2EA	8D AB 02	STA	683
49901	C2ED	B1 F7	LDAIY	247

Address Decml	Hex	Machine Code	Assembly Code Program	
49903	C2EF	29 07	ANDIM	7
49905	C2F1	8D AC 02	STA	684
49908	C2F4	A9 00	LDAIM	0
49910	C2F6	8D AD 02	STA	685
49913	C2F9	A9 20	LDAIM	32
49915	C2FB	8D AE 02	STA	686
49918	C2FE	18	CLC	
49919	C2FF	AD AD 02	LDA	685
49922	C302	6D A7 02	ADC	679
49925	C305	8D AD 02	STA	685
49928	C308	AD AE 02	LDA	686
49931	C30B	6D A9 02	ADC	681
49934	C30E	8D AE 02	STA	686
49937	C311	18	CLC	
49938	C312	AD AD 02	LDA	685
49941	C315	6D AA 02	ADC	682
49944	C318	8D AD 02	STA	685
49947	C31B	AD AE 02	LDA	686
49950	C31E	6D AB 02	ADC	683
49953	C321	8D AE 02	STA	686
49956	C324	18	CLC	
49957	C325	AD AD 02	LDA	685
49960	C328	6D AC 02	ADC	684
49963	C32B	8D AD 02	STA	685
49966	C32E	A9 00	LDAIM	0
49968	C330	6D AE 02	ADC	686
49971	C333	8D AE 02	STA	686
49974	C336	18	CLC	
49975	C337	AD A8 02	LDA	680
49978	C33A	29 07	ANDIM	7
49980	C33C	8D AF 02	STA	687
49983	C33F	A9 07	LDAIM	7
49985	C341	38	SEC	
49986	C342	ED AF 02	SBC	687
49989	C345	8D AF 02	STA	687
49992	C348	18	CLC	
49993	C349	AD AD 02	LDA	685
49996	C34C	85 FB	STAZ	251
49998	C34E	AD AE 02	LDA	686
50001	C351	85 FC	STAZ	252
50003	C353	AD AF 02	LDA	687
50006	C356	18	CLC	
50007	C357	85 02	STAZ	2
50009	C359	A9 01	LDAIM	1
50011	C35B	85 FE	STAZ	254
50013	C35D	A5 02	LDAZ	2
50015	C35F	C9 00	CMPIM	0
50017	C361	F0 07	BEQ	7
50019	C363	18	CLC	
50020	C364	26 FE	ROLZ	254

Address Decml	Hex	Machine Code	Assembly Code Program	
50022	C366	C6 02	DECZ	2
50024	C368	D0 F9	BNE	249
50026	C36A	B1 FB	LDAIY	251
50028	C36C	05 FE	ORAZ	254
50030	C36E	91 FB	STAIY	251
50032	C370	18	CLC	
50033	C371	A9 01	LDAIM	1
50035	C373	65 F7	ADCZ	247
50037	C375	85 F7	STAZ	247
50039	C377	A9 00	LDAIM	0
50041	C379	65 F8	ADCZ	248
50043	C37B	85 F8	STAZ	248
50045	C37D	18	CLC	
50046	C37E	CE F6 02	DEC	758
50049	C381	F0 03	BEQ	3
50051	C383	4C 9C C2	JMP	49820
50054	C386	20 BE C3	JSR	50110
50057	C389	AD F9 02	LDA	761
50060	C38C	49 01	EORIM	1
50062	C38E	F0 03	BEQ	3
50064	C390	4C 9C C2	JMP	49820
50067	C393	60	RTS	
50068	C394	A9 00	LDAIM	0
50070	C396	8D F8 02	STA	760
50073	C399	A5 FD	LDAZ	253
50075	C39B	45 F7	EORZ	247
50077	C39D	85 FC	STAZ	252
50079	C39F	A5 FE	LDAZ	254
50081	C3A1	45 F8	EORZ	248
50083	C3A3	05 FC	ORAZ	252
50085	C3A5	F0 10	BEQ	16
50087	C3A7	A0 00	LDYIM	0
50089	C3A9	18	CLC	
50090	C3AA	A9 01	LDAIM	1
50092	C3AC	65 F7	ADCZ	247
50094	C3AE	85 F7	STAZ	247
50096	C3B0	A9 00	LDAIM	0
50098	C3B2	65 F8	ADCZ	248
50100	C3B4	85 F8	STAZ	248
50102	C3B6	60	RTS	
50103	C3B7	18	CLC	
50104	C3B8	A9 01	LDAIM	1
50106	C3BA	8D F8 02	STA	760
50109	C3BD	60	RTS	
50110	C3BE	A9 00	LDAIM	0
50112	C3C0	8D F9 02	STA	761
50115	C3C3	A9 01	LDAIM	1
50117	C3C5	6D A8 02	ADC	680
50120	C3C8	8D A8 02	STA	680
50123	C3CB	A9 00	LDAIM	0

Address Decml	Hex	Machine Code	Assembly Code Program	
50125	C3CD	6D A9 02	ADC	681
50128	C3D0	8D A9 02	STA	681
50131	C3D3	18	CLC	
50132	C3D4	AD A8 02	LDA	680
50135	C3D7	49 40	EORIM	64
50137	C3D9	85 FA	STAZ	250
50139	C3DB	AD A9 02	LDA	681
50142	C3DE	49 01	EORIM	1
50144	C3E0	05 FA	ORAZ	250
50146	C3E2	F0 07	BEQ	7
50148	C3E4	AD F7 02	LDA	759
50151	C3E7	8D F6 02	STA	758
50154	C3EA	60	RTS	
50155	C3EB	18	CLC	
50156	C3EC	A9 01	LDAIM	1
50158	C3EE	8D F9 02	STA	761
50161	C3F1	60	RTS	

Address Decml	Address Hex	Machine Code	Assembly Code Program	
50600	C5A8	A2 04	LDXIM	4
50602	C5AA	A0 00	LDYIM	0
50604	C5AC	B1 FB	LDAIY	251
50606	C5AE	91 FD	STAIY	253
50608	C5B0	C8	INY	
50609	C5B1	D0 F9	BNE	249
50611	C5B3	E6 FC	INCZ	252
50613	C5B5	E6 FE	INCZ	254
50615	C5B7	CA	DEX	
50616	C5B8	D0 F2	BNE	242
50618	C5BA	60	RTS	

Comments for the High-Resolution subroutine

D.1 Clear High-Resolution memory

Locations 49152-49190: Initialize the registers and put C64 in High-Resolution mode. Loactions 49191-49228: Fill the High-Resolution memory (locations 8192 to 16191) with zeros.

D.2 Color assignment

Locations 49229-49273: this will put a designated color (location 49250) in the High-Resolution screen.

D.3 Find channel numbers

Locations 49274-493000: Here the number of channels is found and accordingly will go to appropriate routine.

D.4 High-Resolution: One channel

Loactions 49301-49318: Initialize the locations and addresses.

Loactions 49319-49349: Here the data to be plotted is multiplied by 200.

$$Y * 200 \dashrightarrow A$$

Locations 49350-49365:

Divide the above result by 255.

$$A / 255 \dashrightarrow B$$

Locations 49366-49389: Substract 199 from the above result (B).

$$199 - B \dashrightarrow C$$

and store the result C into memory for plotting.

D.5 High-Resolution: Two channels

Locations 49390-49403: Initialize the locations and addresses being used.

Loactions 49404-49438: Multiply first data by 100.

Y1 * 100 ---> A1

Locations 49439-49454: Divide A1 by 255.

A1 / 255 ---> B1

Locations 49455-49462: Sustract 100 from B1.

B1 - 100 ---> C1

Locations 49463-49530: Advance the data address and do the same for next data.

Y2 * 100 ---> A2

A2 / 255 ---> B2

B2 - 199 ---> C2

store C2 in the memory for plotting.

D.6 High-resolution: Three channels

Loactions 49555-49569: Initialize the variables.

Locations 49570-49628:

Y1 * 66 ---> A1

A1 / 255 ---> B1

B1 - 66 ---> C1

and store value C1 into the memory for plotting.

Locations 49629-49702:

Y2 * 66 ---> A2

A2 /255 ---> B2

132 - B2 ---> C2

and store C2 for plotting.

Locations 49703-49795:

Y3 * 67 ---> A3

A3 / 255 ---> B3

199 - B3 ---> C3

and store C3 for plotting.

D.7 Plot the adjusted values

Locations 49796-49810: Initialize and create time base X.

Loactions 49811-49832:

int(X/8) * 8 ---> X1

int(Y/8) * 8 ---> Y1

Loactions 49833- 49900:

Y1 * 320 ---> YY

Locations 49901-49919:

Y1 AND 7 ---> Y7

Loactions 49920-49974:

Y7 + YY + X1 + 8192 ---> Addr.

Loactions 49975-50032:

X AND 7 ---> X7

7 - X7 ---> BI

2 ** BI ---> CI

and finally store the CI into the Addr.

Locations 50033-50067: Adjust the address and repeat for the numbers of data per channels.

Locations 50068-50109: Adjust the scale and return a false flag if the last channel.

Locations 50110-50161: Adjust the address and return a false flag if is the last address.

Loactions 50600-50618: This is a utility for plotting routine. This routine will transfer the real data to another block of memory.

Appendix E

HIGH-RESOLUTION SCREEN TO PRINTER ROUTINE

The following is the subroutine to transfer the High-Resolution screen to the printer. The routine will create a graphical character and then will send the character to the printer.

High-Resolution to printer subroutine

Address Decml	Hex	Machine Code	Assembly Code Program	
50280	C468	A9 00	LDAIM	0
50282	C46A	8D BC 02	STA	700
50285	C46D	A9 20	LDAIM	32
50287	C46F	8D BD 02	STA	701
50290	C472	A9 27	LDAIM	39
50292	C474	85 FB	STAZ	251
50294	C476	18	CLC	
50295	C477	A9 00	LDAIM	0
50297	C479	85 FC	STAZ	252
50299	C47B	A9 00	LDAIM	0
50301	C47D	85 FD	STAZ	253
50303	C47F	85 FE	STAZ	254
50305	C481	A0 28	LDYIM	40
50307	C483	18	CLC	
50308	C484	A5 FC	LDAZ	252
50310	C486	65 FD	ADCZ	253
50312	C488	85 FD	STAZ	253
50314	C48A	A9 00	LDAIM	0
50316	C48C	65 FE	ADCZ	254
50318	C48E	85 FE	STAZ	254
50320	C490	88	DEY	
50321	C491	D0 F0	BNE	240
50323	C493	18	CLC	
50324	C494	A5 FB	LDAZ	251
50326	C496	65 FD	ADCZ	253
50328	C498	85 FD	STAZ	253
50330	C49A	A9 00	LDAIM	0
50332	C49C	65 FE	ADCZ	254
50334	C49E	85 FE	STAZ	254
50336	C4A0	A0 03	LDYIM	3
50338	C4A2	18	CLC	
50339	C4A3	26 FD	ROLZ	253
50341	C4A5	26 FE	ROLZ	254
50343	C4A7	88	DEY	
50344	C4A8	D0 F8	BNE	248
50346	C4AA	18	CLC	
50347	C4AB	AD BC 02	LDA	700
50350	C4AE	65 FD	ADCZ	253
50352	C4B0	85 FD	STAZ	253
50354	C4B2	AD BD 02	LDA	701
50357	C4B5	65 FE	ADCZ	254
50359	C4B7	85 FE	STAZ	254
50361	C4B9	A0 00	LDYIM	0
50363	C4BB	B1 FD	LDAIY	253
50365	C4BD	4A	LSRA	

Address Decml	Hex	Machine Code	Assembly Code Program	
50366	C4BE	09 80	ORAIM	128
50368	C4C0	20 D2 FF	JSR	65490
50371	C4C3	A9 00	LDAIM	0
50373	C4C5	91 FD	STAIY	253
50375	C4C7	C8	INY	
50376	C4C8	98	TYA	
50377	C4C9	49 08	EORIM	8
50379	C4CB	D0 EE	BNE	238
50381	C4CD	E6 FC	INCZ	252
50383	C4CF	A5 FC	LDAZ	252
50385	C4D1	49 19	EORIM	25
50387	C4D3	D0 A6	BNE	166
50389	C4D5	A9 0D	LDAIM	13
50391	C4D7	20 D2 FF	JSR	65490
50394	C4DA	C6 FB	DECZ	251
50396	C4DC	10 98	BPL	152
50398	C4DE	60	RTS	

Comments for the printer subroutine

Locations 50280-50303: Initialize the variables and locations to be used in this subroutine.

Locations 50304-50336: Create the graphical character from the High-Resolution screen.

Locations 50337-50398: Send the character to the printer, and check to see if the transfer is completed.

Appendix F

AMPLIFIERS DATA

The amplifiers used in the DAS system were calibrated and tested and the gain for each selected resistor was found. Figure F.1 through F.6 shows the calibration curves for each amplifier and selected gain. The characteristic linear equations for each amplifier are as follow:

TABLE F.1

Amplifiers gain

AMP No.	Expected gain	Actual gain
1	100	101.67
1	500	501.49
1	1000	1000.65
2	100	102.09
2	500	504.34
2	1000	1015.25

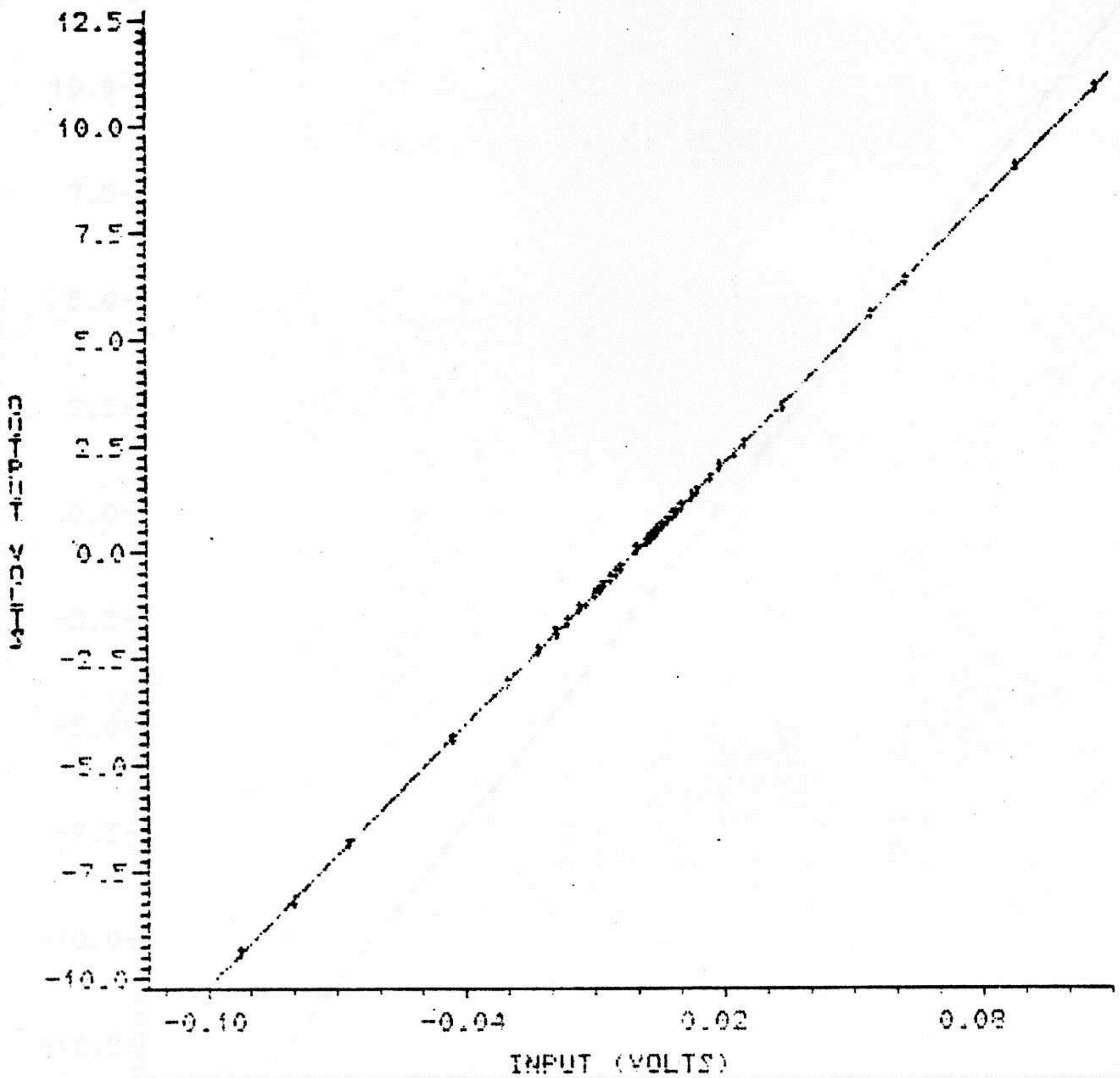


Figure F.1: Calibration curve for amplifier No. 1 (gain=101.67)

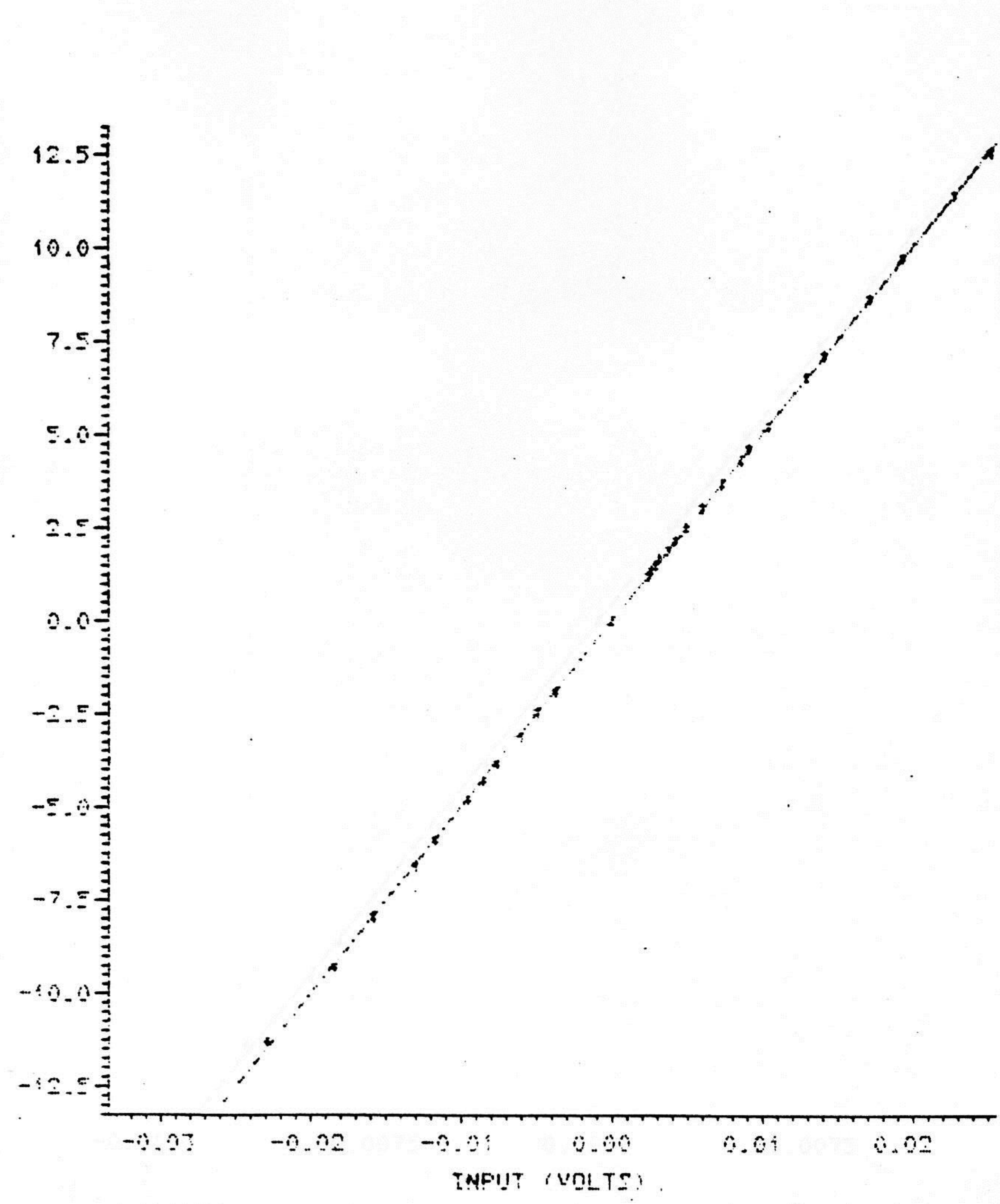


Figure F.2: Calibration curve for amplifier No. 1 (gain=501.49)

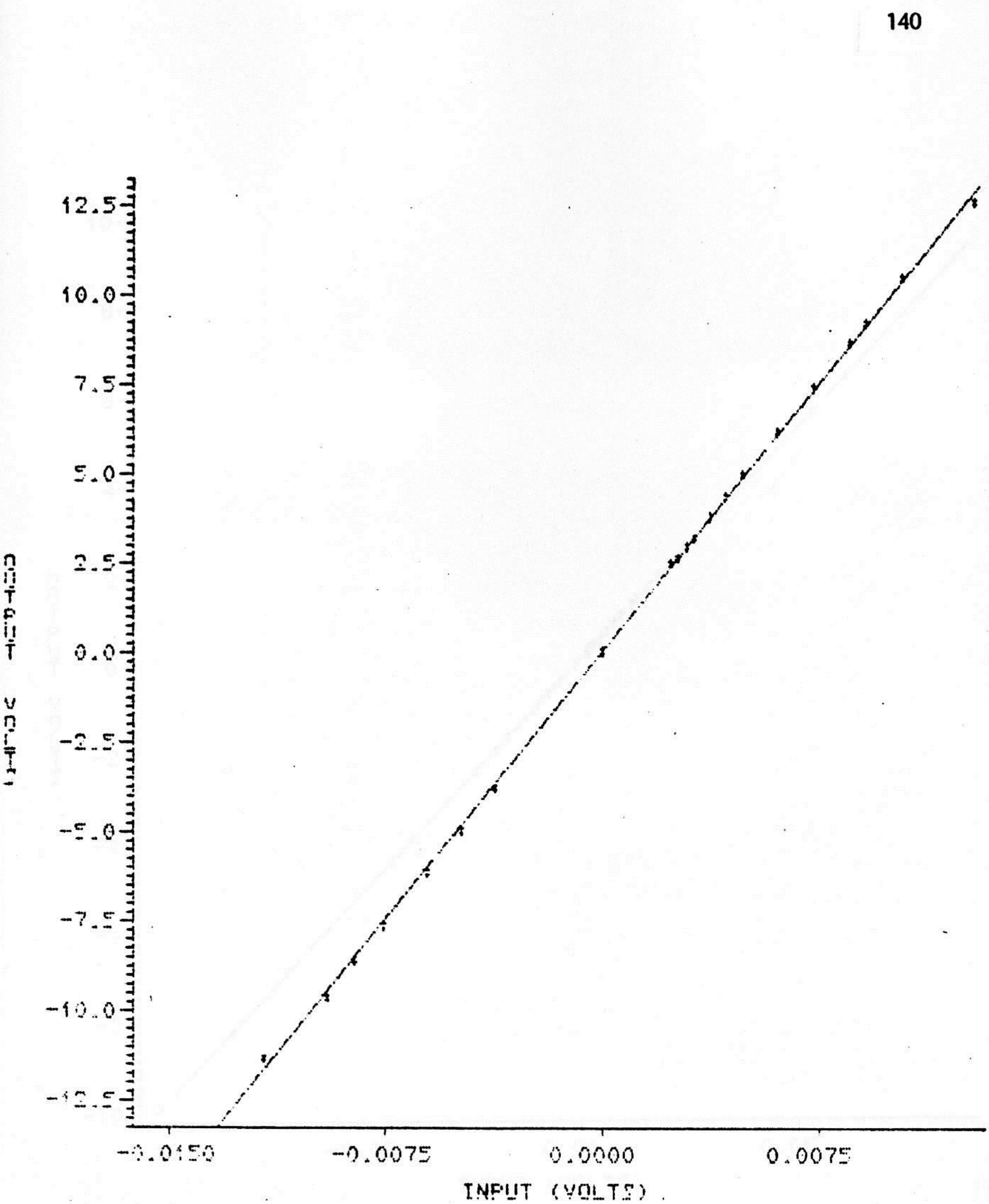


Figure F.3: Calibration curve for amplifier No. 1 (gain=1000.65)

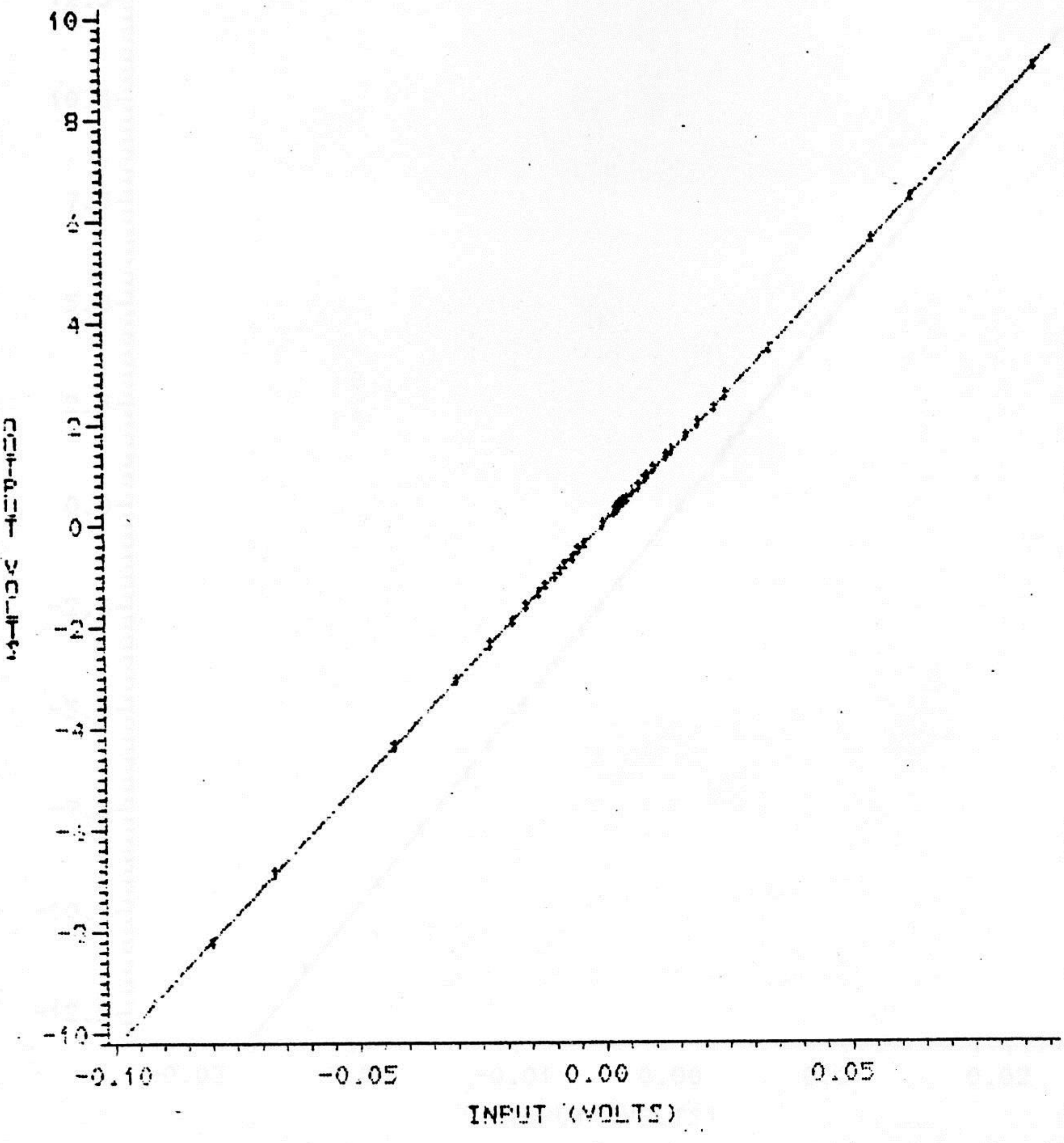


Figure F.4: Calibration curve for amplifier No. 2 (gain=102.09)

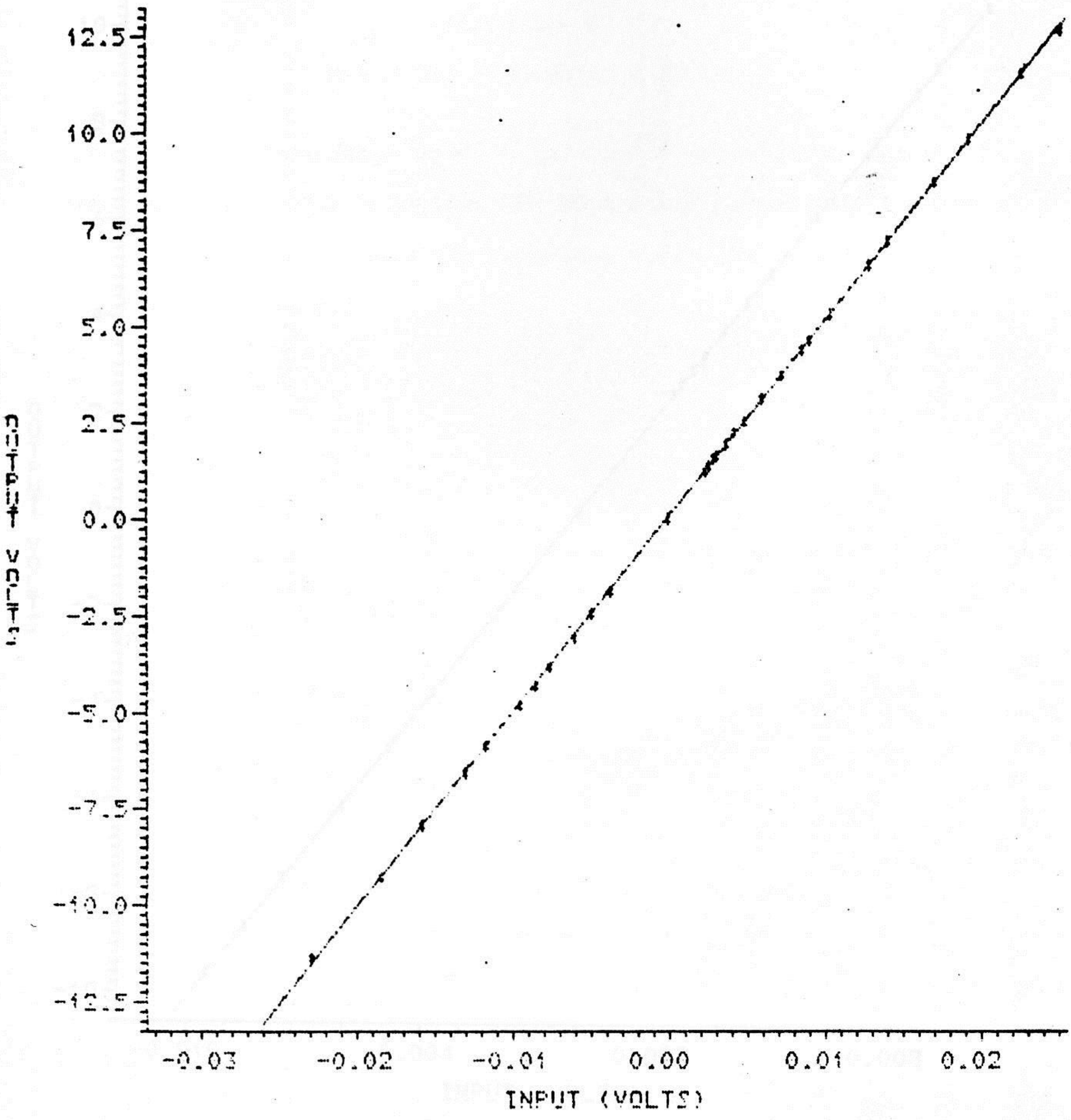


Figure F.5: Calibration curve for amplifier No. 2 (gain=504.34)

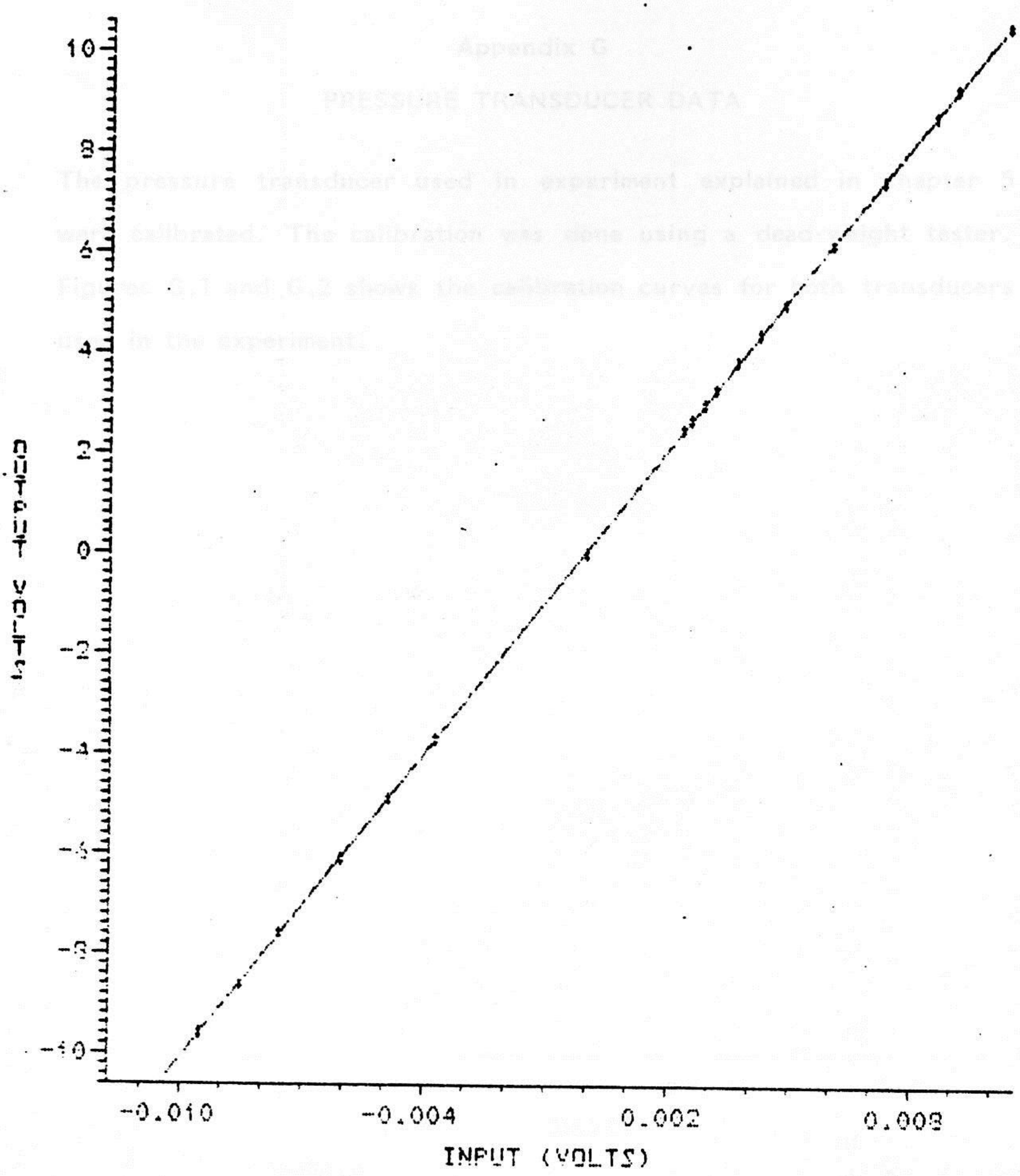


Figure F.6: Calibration curve for amplifier No. 2 (gain= 1015.25)

Appendix G

PRESSURE TRANSDUCER DATA

The pressure transducer used in experiment explained in chapter 5 were calibrated. The calibration was done using a dead-weight tester. Figures G.1 and G.2 shows the calibration curves for both transducers used in the experiment.

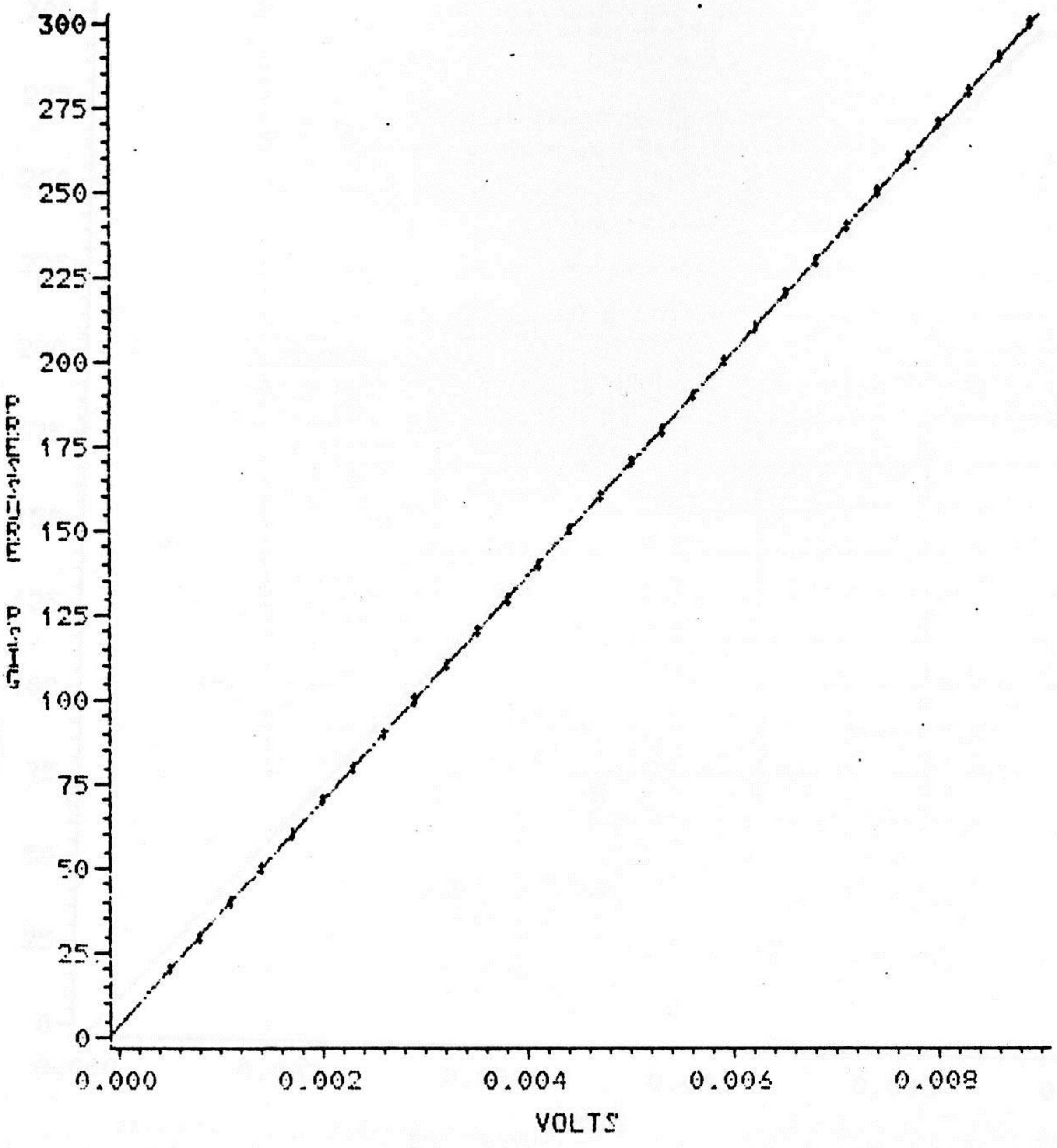


Figure G.1: Calibration curve for pressure transducer No. 1

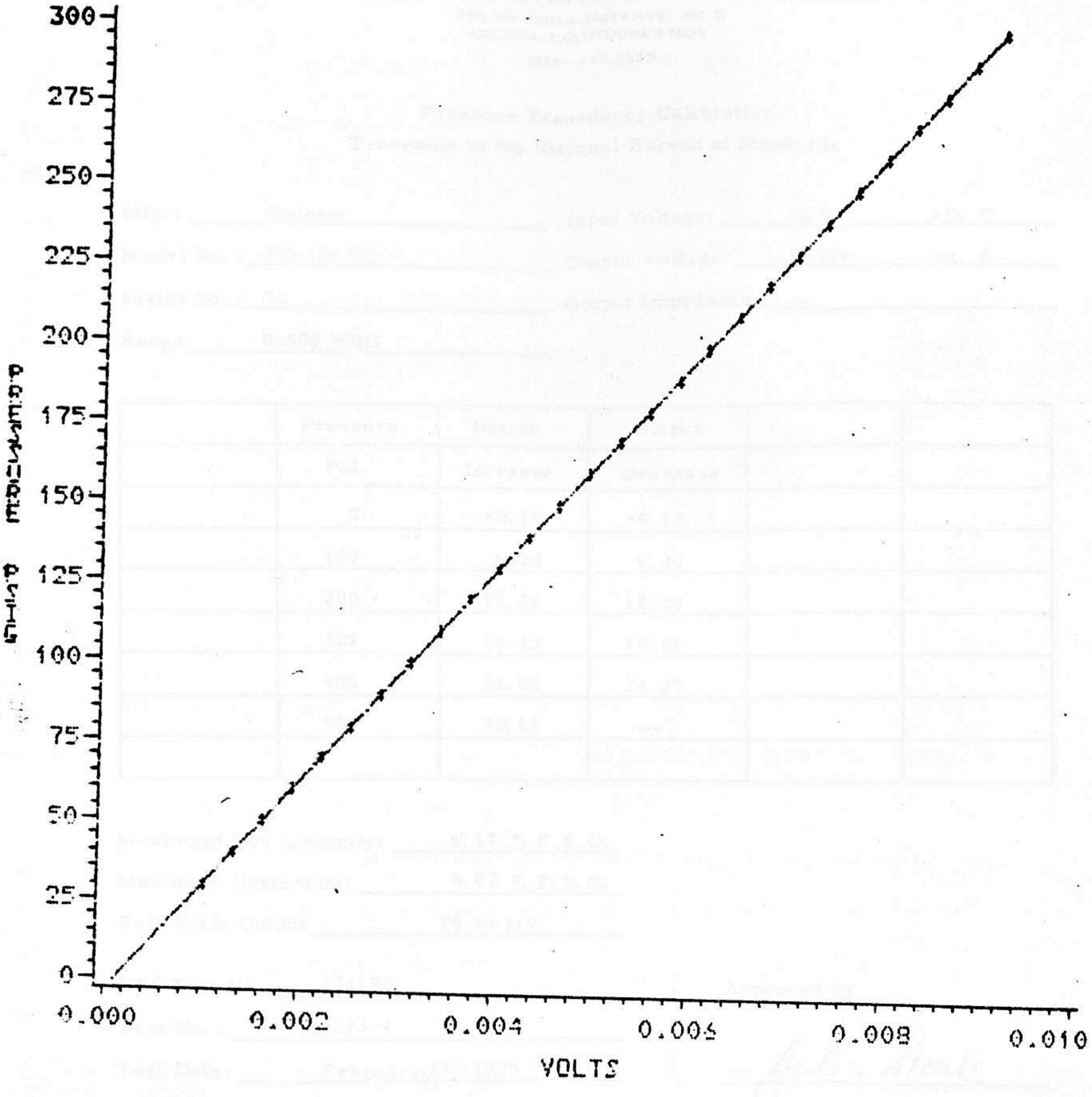


Figure G.2: Calibration curve for pressure transducer No. 2

TEMPERATURE STANDARDS LABORATORY

333 NO. SANTA ANITA AVE., NO. 8
ARCADIA, CALIFORNIA 91006
(213) 445-3227

Pressure Transducer Calibration
Traceable to the National Bureau of Standards

Mfg.: Statham — Input Voltage: 10 V +D, C

Model No.: PG 328 TC — Output Voltage: 30 MV +A, B

Serial No.: 72 — Output Impedance: --

Range: 0-500 PSIG

	Pressure	Output	Output		
	PSI	Increase	Decrease		
	0	+0.19	+0.19		
	100	6.20	6.20		
	200	12.22	12.20		
	300	18.23	18.21		
	400	24.22	24.21		
	500	30.18	--		

Maximum Non Linearity: 0.17 % F.S.O.

Maximum Hysteresis: 0.07 % F.S.O.

Full-Scale Output 29.99 MV

NBS Test No.: 174192

Test No.: 3233-4

Test Date: February 14, 1979

Approved by

John Stork

Appendix H

USER MANUAL FOR COMPRESSOR

DATA ACQUISITION SYSTEM C64 DACQ1

SOFTWARE: COMPRESSOR1

USER'S MANUAL

INTRODUCTION

This manual is intended for a first time user of the Commodore 64-based data acquisition system C64DACQ-1 which has been developed at the Applied High-Tech Laboratory.

The minimum necessary procedure is described in this manual. The user who is interested in technical detail should read:

(1) K. Okamura & K. Aghai-Tabriz,"A Low Cost Data Acquisition System" , BYTE (the small system journal- McGraw-Hill), Vol 10,No.2, February, 1985.

(2) K. Aghai-Tabriz, MS Thesis .

SUMMARY OF SYSTEM

The objective of the system in conjunction with software "compressor1" is to acquire the data necessary to determine the P-V diagram and work of the two-stage compressor used in MEAM 408 Mechanical Engineering Laboratory. The acquired data are shown on a CRT monitor as pressure vs. time for low pressure stage, pressure vs. time for high pressure stage and marking of top dead center of the low pressure stage. Also subsequently shown on the CRT is the corresponding P-V diagram.

When the P-V diagram proves to be vaild, the data can be stored in a disk and/or transmitted to the TRS-80 (Dolve 123) through the coaxial cable. The data stored in the Commodore 64 disk or TRS-80 disk can be further transmitted to the NDSU main frame.

The user has four options to obtain a hardcopy of the P-V diagram:

(1) SAS program - Main Frame

(2) Dot-Matrix Printer - TRS-80

(3) Commodore 64 Dot Matrix Printer

(4) HP XY- recorder with D/A converter
which has been developed in the
Applied High-Tech Laboratory.

The first option will be generally used in MEAM 408 since the data are stored in the main frame and available to the participating students,

each student should be able to calculate the work by numerical integration which was covered in MEAM 107 and 210.

The data transmission procedure is covered in C64DACQ1 "TRANSMISSION1 USER'S MANUAL".

Hardware Connection

(1) Before connecting any equipment to the power source, be sure all power is switched off.

(2) Connect the disk drive and CRT to the C-64.(Fig H.1)

(3) Connect DACQ-1 to the C-64. (Fig H.2)

(4) Connect pressure transducers and the photo sensor to DACQ-1. (Fig H.3)

(5) Switch to ADC.

The entire system is illustrated by a block diagram in Fig H.4.

POWER ON SEQUENCE

The order of switching on devices is important in order to avoid any damage to the equipment.

(1) Check that no disk is in the disk drive;
(2) Turn on CRT;
(3) Turn on disk drive;
(4) Turn on DACQ-1;
(5) Turn on C-64.

Shown on the CRT will be:

```
**** COMMODORE 64 BASIC V ****
64K RAM SYSTEM 39811 BASIC BYTE FREE
READY
-
```

LOADING PROGRAM

(1) Insert disk "Compressor 1" in the disk drive,
close the latch.
(2) Type : LOAD"COMPRESSOR1",8
and press RETURN key (R).
(3) Display will be :
SEARCHING FOR COMPRESSOR1

LOADING

(wait until the screen displays:)

READY

(4) Type : RUN

and press RETURN key.

(5) The screen will black out and the

disk drive will run.

After about 5 sec the menu appears

on the CRT.

(6) Remove disk Compressor1 from the

disk drive.

PRELIMINARY PROCEDURE

Prior to data acquisition, the mechanical and electronic preliminary procedure must be taken.

A- Mechanical Preliminary

Open the relief valve of each pressure transducer. Then open the tank release valve. This will relieve the pressure in each cylinder of the compressor to atmosphere and set the corresponding pressure transducer at the atmospheric pressure, i.e. 0 Psig.

B- Electronic Preliminary

The procedure should start with menu display.

(1) Press B (Bias control)

NOTICE-->>This should be done only once at the beginning of a lab session.

(2) Display :

Low pressure	High pressure
125	207

(3) Gain selectors 1 (Low pressure) and 2 (High pressure) should be set at 1000.

(4) The flashing numerals on display are biases of amplifiers. These values can be adjusted by bias control knobs 1 (low pressure) and 2 (high pressure). Each value should be set between 2 and 5.

(5) Hold down space key.

Menu shows up again.

This completes the preliminary procedure and you are ready to take data.

DATA ACQUISITION PROCEDURE

The procedure starts with Menu.

Press D. (take data in).

The CRT displays:

COMPUTER IN PROCESS

(delay)

PRESS ANY KEY TO CONTINUE

At this point the data are in RAM. Press any key. Menu will be displayed again.

Although you can select any option at this point, the following sequence is highly recommended:

PLOT VS. TIME

Press P. Shown on the CRT are the low pressure vs. time (top), the high pressure vs. time (middle) and the photo sensor output (bottom) which indicates the time marker representing the top dead center of the low pressure cylinder.

Press any key . Back to Menu !

P-V DIAGRAM PLOT

Press V. A series of white dots appear. Wait for a while. These dots will gradually form the P-V diagram for each of the high and low pressure stages. Do not touch any key until the entire diagrams are completed. Compare this plot with a standard plot .

Press any key. Back to Menu !

If the result is satisfactory, proceed to the following; otherwise, go back to the acquisition procedure.

DATA STORING PROCEDURE

Press S.

Display line 1 indicates:

Please insert the data disk.

Insert the appropriate disk on which you intend to save the data. Close the latch.

Display line 2 indicates:

Please enter the name of the file

? -

Type any name less than ten characters and numerals starting with a character.

Press RETURN key

Display shows:

Please Wait!

(delay)

PRESS ANY KEY TO CONTINUE

When you press any key, you go back to Menu!

POWER OFF PROCEDURE

(1) Remove disk form the disk drive;

(2) Turn off the C-64;

(3) Turn off the disk drive and CRT;

(4) Turn off DACQ1.

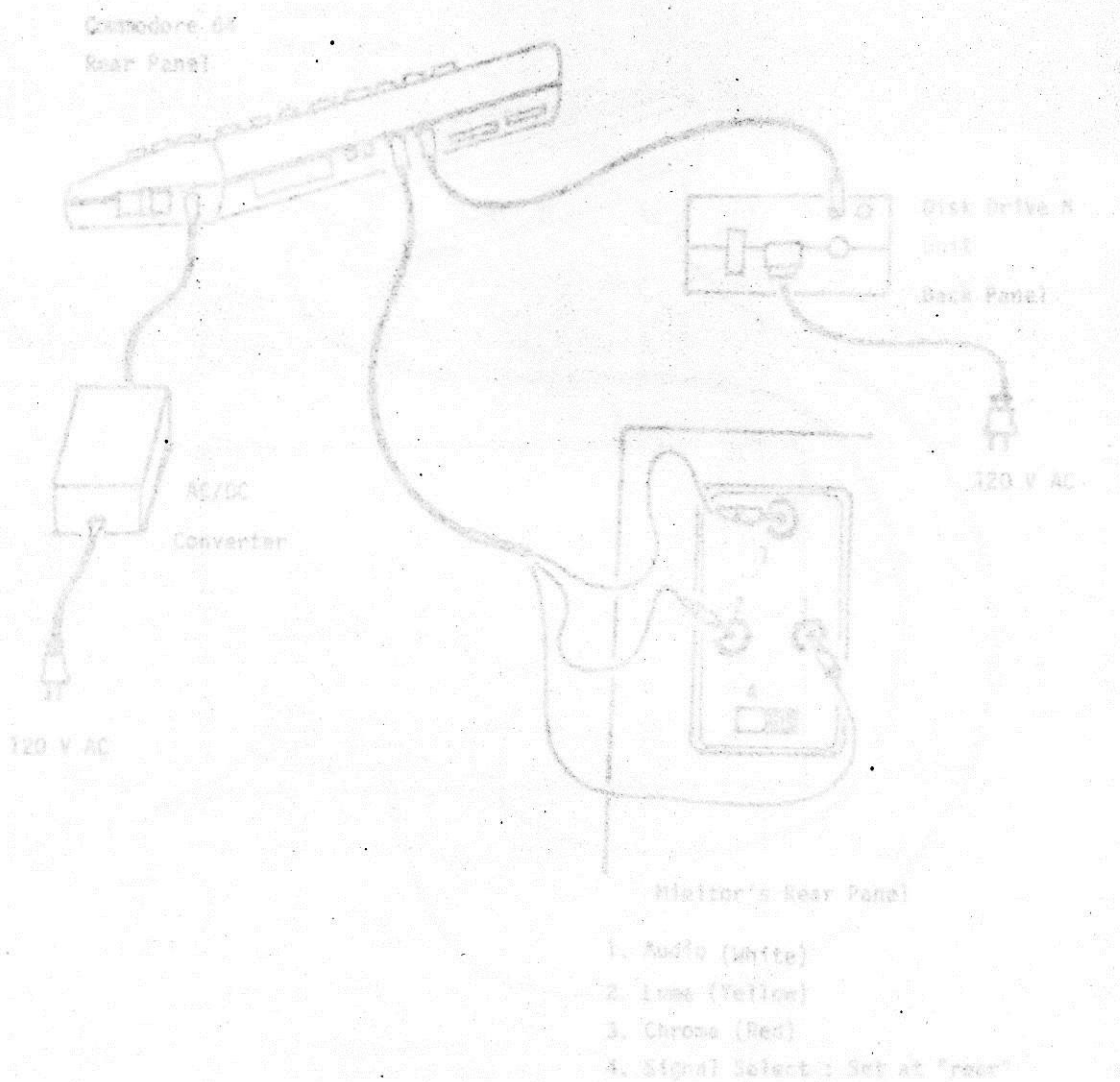


Figure H.1: Commodore 64 and peripheral

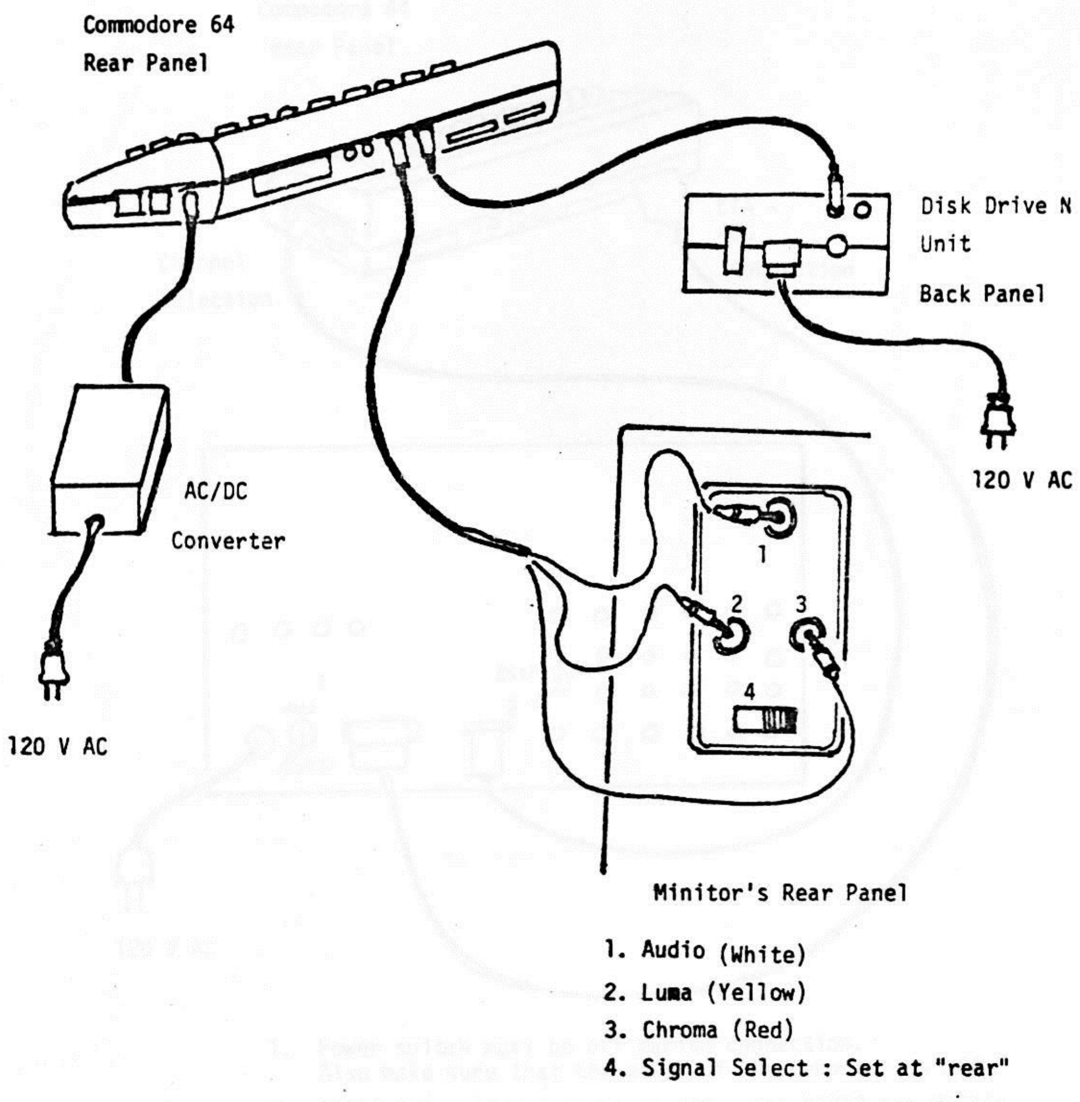
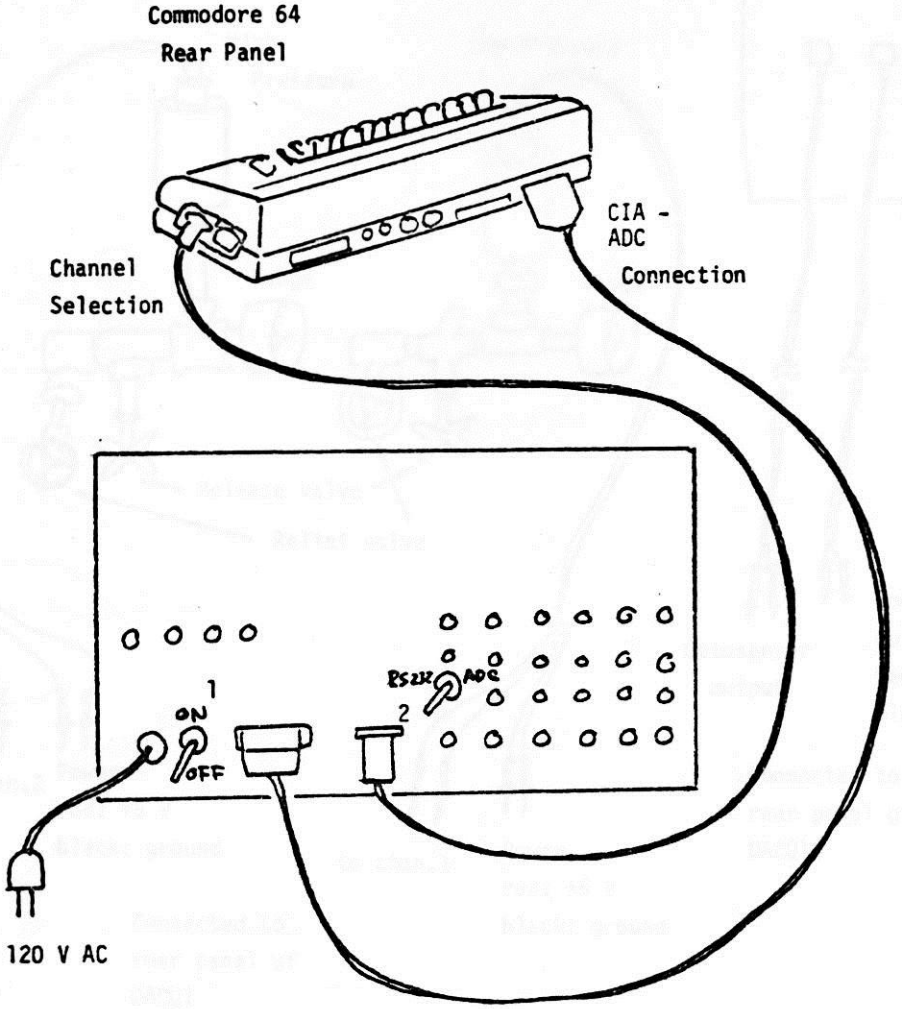


Figure H.1: Commodore 64 and peripheral



1. Power switch must be off during connection.
   Also make sure that the power of the computer is off.
2. RS232-ADC switch must be at ADC. For RS232 see USER's Manual for DACQ1 Data Transmission System.

Figure H.2: Connection for Commodore 64 and DACQ1

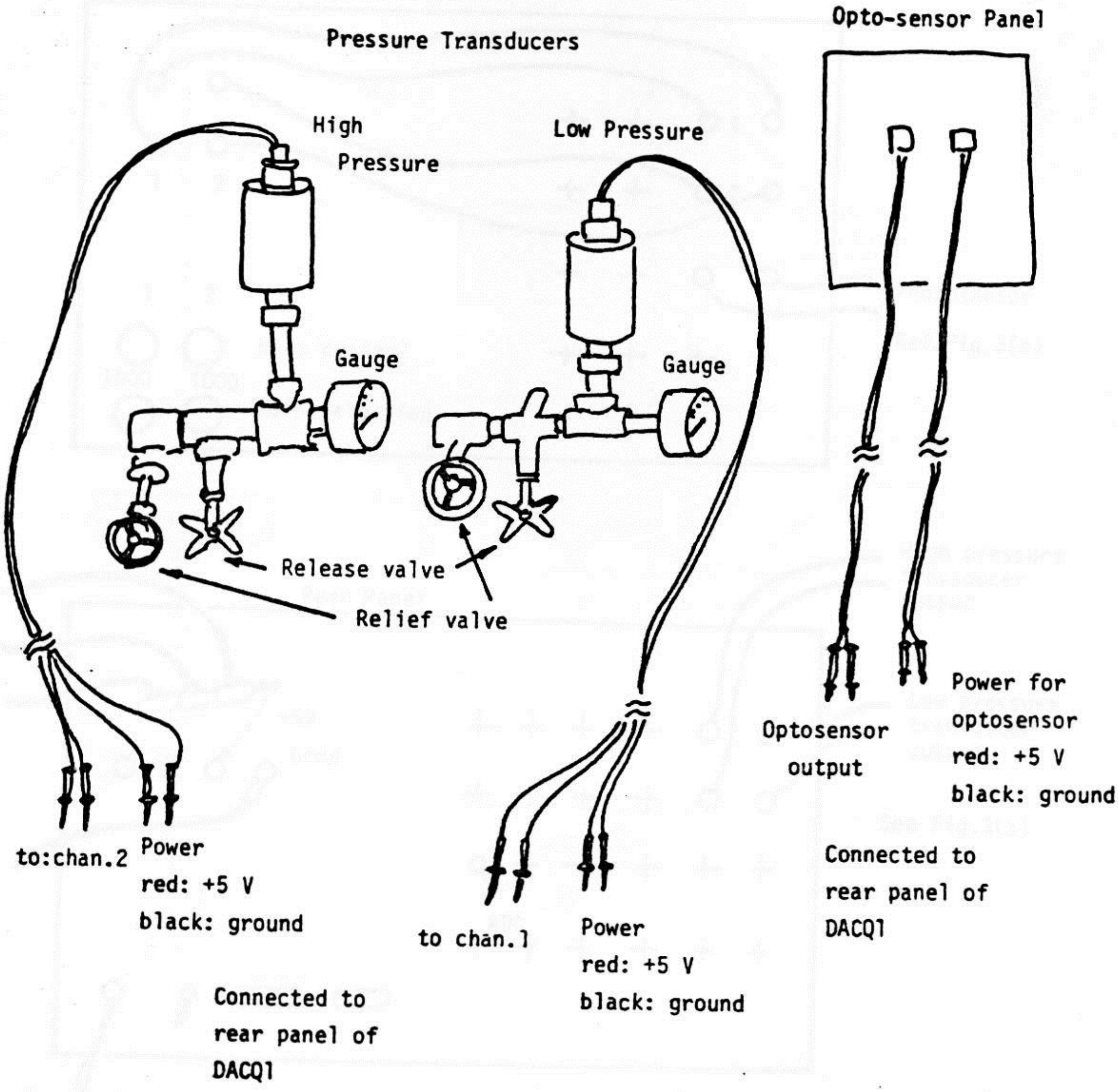


Figure H.3: Pressure Transducer- DACQ1 connection (1)

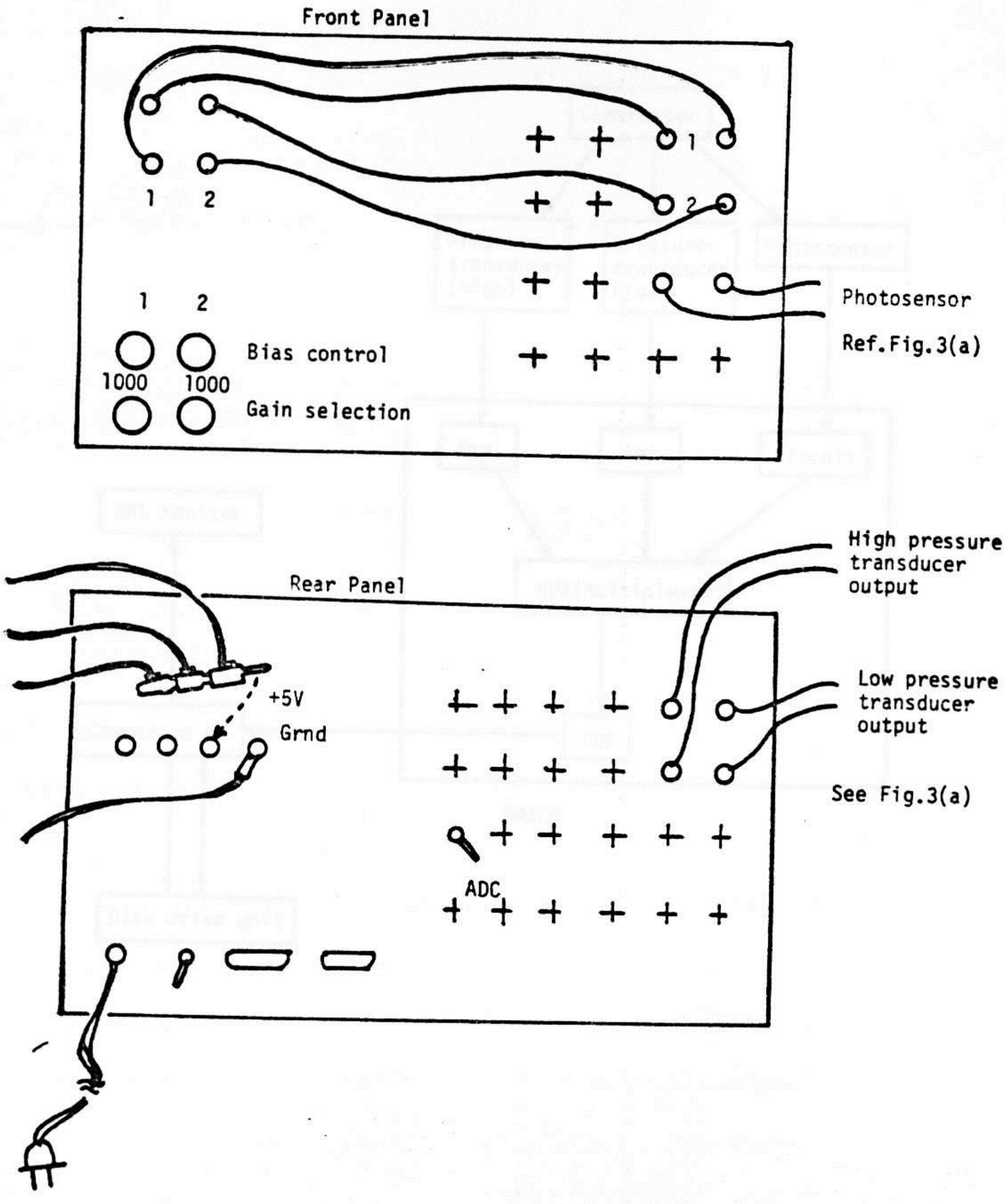


Figure H.4: Pressure Transducer- DACQ1 connection (2)

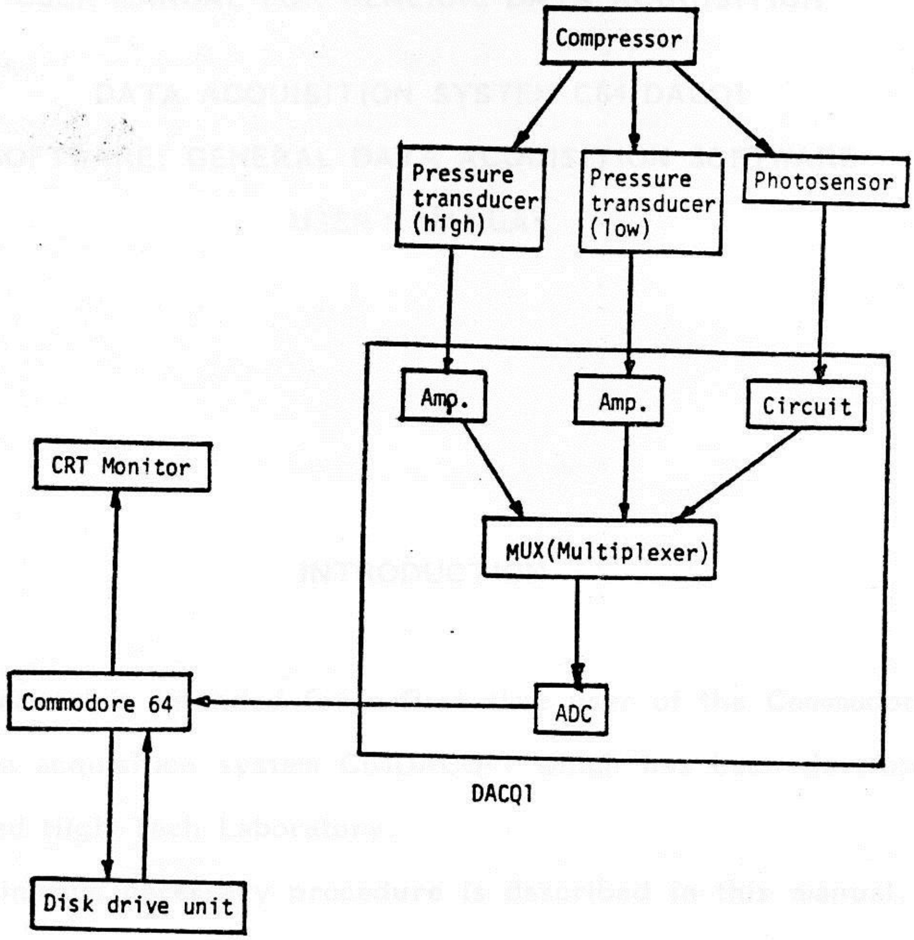


Figure H.5: Signal flow block diagram for DACQ1 and Commodore 64

Appendix I

USER MANUAL FOR GENERAL DATA ACQUISITION

DATA ACQUISITION SYSTEM C64 DACQ1

SOFTWARE: GENERAL DATA ACQUISITION SOFTWARE

USER'S MANUAL

INTRODUCTION

This manual is intended for a first time user of the Commodore 64-based data acquisition system C64DACQ-1 which has been developed at the Applied High-Tech Laboratory.

The minimum necessary procedure is described in this manual. The user who is interested in technical detail should read:

(1) K. Okamura & K. Aghai-Tabriz,"A Low Cost Data Acquisition System" , BYTE (the small system journal- McGraw-Hill), Vol 10,No.2, February, 1985.

(2) K. Aghai-Tabriz, MS Thesis

Hardware Connection

(1) Before connecting any equipment to the power source, be sure all power is switched off.

(2) Connect the disk drive and CRT to the C-64.(Fig H.1)

(3) Connect DACQ-1 to the C-64. (Fig H.2)

(4) Switch to ADC.

POWER ON SEQUENCE

The order of switching on devices is important in order to avoid any damage to the equipment.

(1) Check that no disk is in the disk drive;

(2) Turn on CRT;

(3) Turn on disk drive;

(4) Turn on DACQ-1;

(5) Turn on C-64.

Shown on the CRT will be:

**** COMMODORE 64 BASIC V ****

64K RAM SYSTEM 39811 BASIC BYTE FREE

READY

-

LOADING PROGRAM

(1) Insert disk "DACQ" in the disk drive,

close the latch.

(2) Type : LOAD"DACQ",8

and press RETURN key (R).

(3) Display will be :

SEARCHING FOR DACQ

LOADING

(wait until the screen displays:)

READY

(4) Type : RUN

and press RETURN key.

(5) The first screen will be displayed;

disk drive will run.

After about 50 sec the the screen will display

PRESS ANY KEY TO CONTINUE

on the CRT.

(6) Remove disk DACQ from the disk drive

(7) Press a key .

the CRT will display the program menu.

DATA ACQUISITION PROCEDURE

Figure I.1 displays the main menu for the general data acquisition. The following is a detailed explanation of each option on the menu.

I.1 Take data in

By pressing D on the keyboard the computer will prompt for the number of channel and the number of data per channel(s) desired figure I.2 " ->>> number of channels (1-3)? ". Any number between 1 to 8 could be entered, but since the plotting routine was designed for only 3-channel display the user can not take in 4 channels of data and attempt to plot all 4 channels at once. If such an attempt is made the program will display an error message and will return to the main menu.

The 320 data per channel is the minimum number of data taken per channel. This can be altered by answering Yes to the next prompt. Next the computer prompts "enter a number between (320-****)?" where number **** depends on number of channels selected.

If the answer to the change is "No" then the next screen will appear, figure I.2. Here the sampling rate for the data acquisition can be selected. Default is the maximum sampling rate possible with this soft-

```
                    MENU
                    ------
         take Data in ...........D
          Plot on screen .........P
        Graph on printer .......G
         Transmit data ..........T
         Store data .............S
          Recall old data ........R
          Exit ...................E

       -----------------------------
         Press any key to continue
       -----------------------------
```

Figure I.1: Main menu of DAS software

ware and hardware which is about 4360 data per second. There are some pre-calculated sampling rates available: 1000 ,500 and 100 samples per second. The user can easily change the ML routine to decrease the sampling rate. However, the maximum rate remains at 4360 data per second.

As soon as the sampling rate is chosen the BASIC program will feed the information acquired to the ML routine and control of program is transfered to the ML routine starting location of 50170 (Appendix C contains complete listing of this routine). The display will show " COMPUTER IS IN PROCESS " while it is converting the analog signals into a sequence of 8-bit digital data and storing them in RAM (Random Access Memory). Upon completion of conversion, the CRT displays " PRESS ANY KEY TO CONTINUE ". By pressing a key the program will

return to the main menu. At this point the data aquired has been stored in RAM in time series fashion.

1.2 Plot on screen

To obtain a graphic representation of the data stored, press P (as long as the number of channels selected is 3 or less). The high resolution bit map plotting routine written in machine language will take over the control of the software and in less than a second will display the stored data in a time series. This can help to check the validity of the data immediately at the experimental site. For further and more complex analysis the data can be transmitted to a larger computer.

To display the same plot, one can use a BASIC or BASIC-ML-mix (see [16] and [7]). BASIC is easier and more flexible to program but slow. BASIC plotting subroutines were tested against the ML routine. To display the same number of data points the BASIC program took approximately two minutes, the BASIC-ML-mix subroutine took 20 seconds and the ML routine took less than a second. The display continues to show the same plot until any key is pressed which will return to the main menu.

```
->>>Number of channels(1-3)?
?

->>>Number of data per channel is 320.

change(Y/N)? Y

enter a number between (320-****)
?
```

Figure 1.2: Sub-menu for channel selection

1.3 Graph on printer

The plot created on a high-resolution screen can be plotted on C-64's, MPS801 printer. Although the quality and resolution of this dot matrix printer is limited, it is satisfactory for situations. For better resolution and higher quality graphic printer or plotter the data can be transmitted to the main frame computer.

The plotting routine is written in machine language. This increases the speed of plotting. For a complete hard copy of plot this routine took about two minutes while for the same task a BASIC routine took about eight minutes (see Appendix for complete listing of this routine).

```
Select Sampling rate
---------------------

 1- Defualt .............
2- 1000 sample per second
3- 500  sample per second
4- 100  sample per second
```

Figure I.3: Sub-menu for sampling rate selection

I.4 Transmit data

The C-64 has a built-in RS-232 interface for serial data communication with another device with an RS-232 port. Although C-64 can be used for data processing to some extent, it may be more efficient to transfer the data to another computer which has more software support for scientific purposes. In other words, the inexpensive C-64 can be used as data acquisition terminal and a more expensive computer can be used as a central processing station.

By pressing T on the keyboard the computer gives the instruction : "Switch to RS-232 position and press any key to transmit" (figure I.4). Switch SW1 (figure 3.4) to the +5V position and press a key. The CRT will display the data as it is being transmitted from C-64. At the end of transmission, the CRT will display an instruction shown on figure I.5:"switch to ADC and press any key to continue." Switch SW1 back to the previous position so that the CIA is connected to the ADC. Transmission to main frame is explained in chapter 6.

Switch to RS-232

and

press any key to transmit

Figure I.4: Sub-menu for data transmission

Switch to ADC position

and

press any key to continue

Figure I.5: Sub-menu for data transmission

I.5 Store data

The main BASIC program has a feature to save the data on a disk. By pressing S on the keyboard the computer will prompt (figure I.6): "please enter the name of the file?" type a string up to 16 characters and numerals starting with a character. After typing the name of the file RETURN key must be pressed. The data will be stored in a se-

quential filing format. The data stored can be retrived later for reviewing or transmission to another computer.

please insert the data disk!!

please enter the name of the file?

press any key to continue

Figure 1.6: Sub-menu for data storage

1.6 Recall data

By pressing R on the keyboard the computer will prompt figure 1.6: "please enter the name of the file?" After typing the name press the RETURN key. A sign "please wait" appears on the CRT. When the tranfer of data from disk to RAM is done this sign will change to press any key to continue. By pressing a key the program will go back to the main menu.

POWER OFF PROCEDURE

(1) Remove disk form the disk drive;

(2) Turn off the C-64;

(3) Turn off the disk drive and CRT;

(4) Turn off DACQ1.

Appendix J

USER MANUAL FOR TRANSMISSION

The program for transmission is listed in this appendix. This program was made only for transmission of the compressor data to main frame. the program can easily be changed to transmit any data file. Here is the step by step explaination for transmission compressor data to main frame via telephone line.

Preliminary Connections

(1)- Connect the modem to the Commodore 64 user port;

(2)- Turn on the disk drive;

(3)- Turn on the CRT;

(4)- Turn on the C64;

(5)- Make sure the switch on modem is set to "O".

Loading the program

(1)- Insert disk "TRANSMISSION 1" in the

disk drive, close the latch;

(2)- Type: LOAD"TRANSMISSION1",8

and press RETURN key

(3)- Display will show:

SEARCHING FOR TRANSMISSION1

LOADING

(wait until the screen displays)

READY.

(4)- Type: RUN

and press RETURN key

(5)- After about 10 seconds display will

clear.

Calling NDSU main frame

(1)- Dial 8661

(2)- wait for the computer tune.

(3)- as the tune is heared, disconnect

the telephone from the hand set and connect

the wire to the modem.

(4)- press RETURN key few times.

(5)- display will show

ENTER CLASS

(6)- Type: 1

and press RETURN key.

(7)- next enter the user number

V xxxxx ; where xxxxx is user number.

(8)- enter the password.

(9)- After a few messages the CRT will display:

READY.

Transmit the data file

(1)- Put the disk containing data file into the disk drive.

(2)- type: INPUT

and press RETURN key.

(3)- the CRT will display

00010

(4)- next press F1 key on the keyboard.

CRT will display

NAME OF THE FILE?

type in the file name, and press RETURN.

(5)- next, CRT will display

Number of data to be transmitted?

type in number 320. and press RETURN

(6)- The disk drive starts and the data transmitted will be displayed on CRT.

(7)- after all data are transmitted CRT will display.

--->> TRANSMISSION COMPLETED <<---

(8)- press RETURN key to get the VSPC in READY mode.

(9)- now type: SAVE NAME

(10)- at this point the data are successfully transmitted and save in VSPC.

Any program can be written to manipulate this data.

Procedure log off

(1)- type : OFF and press RETURN

(2)- the CRT will display the connection time , CPU time and will log off.

(3)- connect the telephone and hang-up.

BIBLIOGRAPHY

[1]--Okamura, K. and Aghai-Tabriz K. A Low Cost Data Acquisition System BYTE, New hampshire: McGraw-Hill, FEB. 1985.

[2]-- TRS-80 Reference Manual, Texas, Tandy Corporation, 1978.

[3]--Lenk, John, D. Handbook of microcomputer-based instrumentation and controls. N.J.: Prentice-Hall, INC, 1984.

[4]-- Data Acquisition and Conversion , Analog Device Inc, MA, 1982.

[5]-- Linear Databook, National Semiconductor Corporation, 1980.

[6]--Hallmark, Clayton, L. The Master IC Cookbook, Blue Ridge Summit,PA, TAB Books Inc., 1980.

[7]-- Commodore 64 Programmer's Reference Guide. Commodore Business Machine, Inc., 1984

[8]--Williams, Arthur, B. Designer's Handbook of Integrated Circuits. N.Y.: McGraw-Hill, 1984.

[9]--Leemon, Sheldon. Mapping The Commodore 64, North Carolina, COMPUTE Publications, Inc., 1984.

[10]-- Data Acquisition and Conversion Handbook. California, INTERSIL, California, INTERSIL Inc,:1980.

[11]-- Reference Manual for 1541 Commodore Disk Drive. Commodore Business Machine, Inc., 1983.

[12]--Okamura K., MEAM 306 Laboratory Notes. NDSU, 1982

[13]-- SAS/Graph User's Guide. SAS Institute Inc., North Carolina, 1981.

[14]-- SAS User's Guide. SAS Institute Inc., North Carolina, 1981.

[15]--Pish, Robert, H. A New-Generation Cylinder Performance Indicator. Mechanical Engineering Magazine. Dec 1984,pp 61-69

[16]-- 64 Sound and Graphics, Compute Books Publications, North Carolina, 1983.

[17]--Zaks, Rodnay. Programming the 6502. Berkeley, California, SYBEX, 1980.

[18]--Zaks, Rodnay. <u>6502 Applications</u>. Berkeley, California, SYBEX, 1979.

[19]--Leventhal, Lance, A. <u>6502 Assembly Language Programming</u>. Berkeley, California, OSBORN/McGraw-Hill, 1979.

[20]--Scanlon, Leo, J. <u>6502 Software Design</u>. Indiana: Howard W. Sams & Co.,Inc.,1980.